

15th Symposium on Languages, Applications and Technologies

SLATE 2026, June 25–26, 2026, Lisbon, Portugal

Edited by

Fernando Batista

Eugénio Ribeiro

Ricardo Ribeiro

André L. Santos



Editors

Fernando Batista

Instituto Universitário de Lisboa (ISCTE-IUL), Portugal
INESC-ID Lisboa, Portugal
fernando.batista@iscte-iul.pt

Eugénio Ribeiro

Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, Lisboa, Portugal
INESC-ID Lisboa, Portugal
eugenio.ribeiro@iscte-iul.pt

Ricardo Ribeiro

Instituto Universitário de Lisboa (ISCTE-IUL), Portugal
INESC-ID Lisboa, Portugal
ricardo.ribeiro@iscte-iul.pt

André L. Santos

Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, Lisboa, Portugal
andre.santos@iscte-iul.pt

ACM Classification 2012

Computing methodologies → Natural language processing; Information systems → World Wide Web;
Theory of computation → Formal languages and automata theory

ISBN 978-3-95977-440-6

Published online and open access by

Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at <https://www.dagstuhl.de/dagpub/978-3-95977-440-6>.

Publication date

August, 2026

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists all publications of this volume in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <https://portal.dnb.de>.

License

This work is licensed under a Creative Commons Attribution 4.0 International license (CC-BY 4.0):
<https://creativecommons.org/licenses/by/4.0/legalcode>.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/OASlcs.SLATE.2026.0

ISBN 978-3-95977-440-6

ISSN 1868-8969

<https://www.dagstuhl.de/oasics>

OASlcs – OpenAccess Series in Informatics

OASlcs is a series of high-quality conference proceedings across all fields in informatics. OASlcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

Editorial Board

- Daniel Cremers (TU München, Germany)
- Barbara Hammer (Universität Bielefeld, Germany)
- Marc Langheinrich (Università della Svizzera Italiana – Lugano, Switzerland)
- Dorothea Wagner (*Editor-in-Chief*, Karlsruher Institut für Technologie, Germany)

ISSN 1868-8969

<https://www.dagstuhl.de/oasics>

■ Contents

Preface	
<i>Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos</i>	0:vii
List of Authors	
.....	0:vii
Committees	
.....	0:xi

Papers

Optimising Retrieval for Linguistic Question-Answering in European Portuguese: A Benchmark on Ciberdúvidas Da Língua Portuguesa	
<i>Pedro Moura, Inês Gama, Fernando Batista, and António Lopes</i>	1:1–1:17
Evaluating the Prosodic Diversity of TTS Models for L2 Prosody Assessment	
<i>Mariana Julião, Alberto Abad, and Helena Moniz</i>	2:1–2:13
Cross-Language Text Readability Assessment: Leveraging Multilingual Models for Improved Performance in CEFR-Level Classification	
<i>Eugénio Ribeiro and Jorge Baptista</i>	3:1–3:13
BERT-Based Classification of Cross-Linguistic MCQs According to the Bloom Taxonomy	
<i>João Francisco Botas, Ana Rita Peixoto, and Eugénio Ribeiro</i>	4:1–4:17
Inter-Sentential Relations in Enhanced Lexicalized Meaning Representation (E-LMR)	
<i>Jorge Baptista</i>	5:1–5:18
A Topic–Word Graph for Topic Modeling Visualization	
<i>Ana Rita Peixoto</i>	6:1–6:11
Polarizations in Static Word Embeddings: Investigating Bias in Marginalized Dialects of Portuguese	
<i>Raimundo Juracy Campos Ferro Junior, Eugénio Ribeiro, and Leonardo Sampaio Rocha</i>	7:1–7:15
Gender Bias Evaluation in English-Portuguese Automated Translation Outputs	
<i>Xiaolan Xu and Sara Mendes</i>	8:1–8:14
Cost-Effective Hate Speech Detection in Portuguese Using Lightweight LLMs	
<i>Mauro Cardoso, Eugénio Ribeiro, and Fernando Batista</i>	9:1–9:16
Assessing LLMs for Culturally Sensitive Communication in European Portuguese	
<i>Alice Vieira, Raquel Amaro, and Lyndon Nixon</i>	10:1–10:14
Developing an AI-Based Application for Linguistic Analysis of Deaf Learners’ L2 Writing	
<i>Aryane Santos Nogueira and Ana Célia Ribeiro Bizigato Portes</i>	11:1–11:10

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

OntoExpand: SPARQL-Based Ontology Expansion and Reasoning <i>Vitor Lelis N. Leite and José Carlos L. Ramalho</i>	12:1–12:15
Deductive Databases and Logic Programming <i>Dietmar Seipel and Salvador Abreu</i>	13:1–13:12
An OWL/RDF Ontology for Declarative User Interface Generation <i>Daniel Theisges da Costa, Júlio Costa, Paulo Jorge Teixeira Matos, Ramon Gallinad Corti, Ramon Vilanova Arbos, and Maria João Varanda Pereira</i>	14:1–14:11
A Silent Multi-Agent Recommendation Engine with Graph-Augmented Memory for Immersive B2B E-Commerce <i>Davi Silva Eleuterio, Pedro Filipe Oliveira, and Paulo Matos</i>	15:1–15:14
Coordinating Multi-Assistant Interaction in Smart Homes: A Front-End Delegation Approach <i>Pedro Carneiro, Inês Águia, Nuno Almeida, and António Teixeira</i>	16:1–16:16
From Natural Language to Structured Queries: An LLM-Powered Chatbot for a Smart Tourism Observatory <i>Rodrigo Bravo Simões, Adriano Lopes, and Fernando Brito e Abreu</i>	17:1–17:11
Automating the Classification of Smart Tourism Tools with LLMs: A Taxonomy-Driven Approach <i>Diogo Cosme, Fernando Batista, António Galvão, and Fernando Brito e Abreu</i> ...	18:1–18:11
Lazarus: Where Compilers Meet Education <i>Rui Gonçalves, Pedro Rangel Henriques, and José Carlos Ramalho</i>	19:1–19:18
Out of the Loop No More: Online Abstract Debugging with Loop Stepping <i>João Alvim, Maarten Steevens, and Christophe Scholliers</i>	20:1–20:10
RI: A DSL for Software Development Project Continuous Configuration <i>Cássio Ritse Machado Dos Santos Silva, Maria João Varanda Pereira, Paulo Alexandre Vara Alves, and Gustavo Jansen de Souza Santos</i>	21:1–21:14

Preface

This volume contains the proceedings of the 15th Symposium on Languages, Applications and Technologies, SLATE 2026, held at Iscte – Instituto Universitário de Lisboa, Lisbon, Portugal, on June 25 and 26, 2026.

Since its inception, SLATE has provided a forum for the discussion of languages in their many forms and uses. Language is central to communication, representation, reasoning, and computation, and the symposium reflects this diversity through its three usual tracks: Human-Human Languages (HHL), Human-Computer Languages (HCL), and Computer-Computer Languages (CCL). These tracks bring together contributions on natural language processing, programming languages, language design and implementation, mark-up and interchange formats, protocols, and the many technologies that support the processing, analysis, and generation of languages.

The 2026 edition continues this tradition by gathering researchers, practitioners, and educators working at the intersection of languages, applications, and technologies. The papers included in these proceedings address both theoretical and practical challenges, ranging from language modelling and linguistic resources to programming language techniques, language-based tools, and applications in emerging technological contexts. Together, they illustrate the continued vitality of language-centred research and its relevance in a rapidly evolving digital landscape.

This year's edition featured two distinguished keynote speakers. Alcides Fonseca, from LASIGE and *Faculdade de Ciências da Universidade de Lisboa*, presented *Æon: A Programming Language with Guardrails*, discussing how advanced type systems, program synthesis, and artificial intelligence techniques can contribute to flexible programming languages. João Magalhães, from NOVA LINC and *Universidade NOVA de Lisboa*, presented *AMALIA: An Open Source Large Language Model for European Portuguese*, addressing the development and evaluation of open language models specifically designed for European Portuguese.

All submitted papers underwent a rigorous peer review process, leading to the acceptance of 21 papers, 3 of which are short papers. We are grateful to the members of the Programme Committee and to the external reviewers for their careful evaluations, constructive comments, and important role in selecting the final contributions. Their work was essential to ensuring the scientific quality of the symposium.

We would also like to thank all authors, affiliated with institutions from six different countries, for sharing their current research, challenges, and results with the SLATE community. Our sincere appreciation goes to the keynote speakers for accepting our invitation and enriching the programme. We also thank the organising committee, institutional partners, and local support teams at Iscte-IUL and ISTAR-Iscte for their dedication and effort in preparing this edition.

We hope that these proceedings offer a stimulating reading experience and inspire further research, collaboration, and dialogue across the communities concerned with languages, applications, and technologies.

The SLATE 2026 Chairs

F. Batista, E. Ribeiro, R. Ribeiro, and A. L. Santos

Iscte-IUL, Lisbon, Portugal

■ List of Authors

Alberto Abad  (2)

HLT, INESC-ID, Lisbon, Portugal; Instituto Superior Técnico, Lisbon, Portugal

Fernando Brito e Abreu  (18)

ISTAR, Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

Salvador Abreu  (13)

NOVA LINC3, University of Évora, Portugal

Nuno Almeida  (16)

IEETA, DETI, University of Aveiro, Portugal; LASI – Intelligent System Associate Laboratory, Guimarães, Portugal

Paulo Alexandre Vara Alves  (21)

Research Centre in Digitalization and Intelligent Robotics (CeDRI), Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, Portugal

João Alvim  (20)

University of Minho, Braga, Portugal

Raquel Amaro  (10)

CLUNL – Research Center for Linguistics at NOVA University Lisbon, NOVA University Lisbon, Portugal

Jorge Baptista  (3, 5)

INESC-ID Lisboa, Portugal; Faculdade de Ciências Humanas e Sociais, Universidade do Algarve, Faro, Portugal

Fernando Batista  (1, 9, 18)

ISCTE – Instituto Universitário de Lisboa, Portugal; INESC-ID Lisboa, Portugal

João Francisco Botas  (4)

Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, Lisboa, Portugal

Rodrigo Bravo Simões  (17)

Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

Fernando Brito e Abreu  (17)

Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

Mauro Cardoso  (9)

Instituto Universitário de Lisboa (ISCTE-IUL), Portugal; INESC-ID Lisboa, Portugal

Pedro Carneiro  (16)

IEETA, DETI, University of Aveiro, Portugal; LASI – Intelligent System Associate Laboratory, Guimarães, Portugal

Ramon Gallinad Corti  (14)

Dept. de Telecomunicació i Enginyeria de Sistemes, Escola d'Enginyeria, Universitat Autònoma de Barcelona, Spain

Diogo Cosme  (18)

ISTAR, Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

Júlio Costa  (14)

Research Centre in Digitalization and Intelligent Robotics (CeDRI), Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, Portugal

Daniel Theisges da Costa (14)

Research Centre in Digitalization and Intelligent Robotics (CeDRI), Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, Portugal

Gustavo Jansen de Souza Santos  (21)

Universidade Tecnológica Federal do Paraná (UTFPR), Dois Vizinhos, Brazil

Cássio Ritse Machado Dos Santos Silva (21)

Instituto Politécnico de Bragança, Portugal

Davi Silva Eleuterio  (15)

CeDRI, SusTEC, Instituto Politécnico de Bragança, Portugal

Raimundo Juracy Campos Ferro Junior  (7)

Universidade Federal do Ceará, Fortaleza, Brasil; ISTAR, Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

António Galvão  (18)

CENSE, School of Science and Technology, NOVA University Lisbon, Caparica, Portugal

Inês Gama  (1)

ISCTE - Instituto Universitário de Lisboa, Portugal

Rui Gonçalves  (19)

ALGORITMI Research Centre / LASI, Dept. of Informatics, University of Minho, Braga, Portugal

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos



OpenAccess Series in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Pedro Rangel Henriques  (19)
ALGORITMI Research Centre / LASI, Dept. of
Informatics, University of Minho, Braga,
Portugal

Mariana Julião  (2)
HLT, INESC-ID, Lisbon, Portugal

Vitor Lelis N. Leite  (12)
University of Minho, Braga, Portugal

Adriano Lopes  (17)
Instituto Universitário de Lisboa (Iscte-IUL),
Portugal

António Lopes  (1)
ISCTE - Instituto Universitário de Lisboa,
Portugal; ISTAR, Lisboa, Portugal

Paulo Matos  (15)
CeDRI, SusTEC, Instituto Politécnico de
Bragança, Portugal

Paulo Jorge Teixeira Matos  (14)
Research Centre in Digitalization and Intelligent
Robotics (CeDRI), Laboratório Associado para
a Sustentabilidade e Tecnologia em Regiões de
Montanha (SusTEC), Instituto Politécnico de
Bragança, Portugal

Sara Mendes  (8)
Center of Linguistics of the University of Lisbon
and Faculty of Arts and Humanities, University
of Lisbon, Portugal

Helena Moniz  (2)
HLT, INESC-ID, Lisbon, Portugal; Faculdade de
Letras da Universidade de Lisboa, Portugal;
CLUL, Lisbon, Portugal

Pedro Moura  (1)
ISCTE – Instituto Universitário de Lisboa,
Portugal

Lyndon Nixon  (10)
Storypact GmbH, Vienna, Austria

Aryane Santos Nogueira  (11)
UNICAMP, Campinas-SP, Brasil

Pedro Filipe Oliveira  (15)
CeDRI, SusTEC, Instituto Politécnico de
Bragança, Portugal

Ana Rita Peixoto  (4, 6)
Instituto Universitário de Lisboa (ISCTE-IUL),
ISTAR, Lisboa, Portugal

Maria João Varanda Pereira  (14, 21)
Research Centre in Digitalization and Intelligent
Robotics (CeDRI), Laboratório Associado para
a Sustentabilidade e Tecnologia em Regiões de
Montanha (SusTEC), Instituto Politécnico de
Bragança, Portugal

Ana Célia Ribeiro Bizigato Portes  (11)
UNICAMP, Campinas-SP, Brasil; FATEC, Mogi
Mirim-SP, Brasil

José Carlos Ramalho  (19)
ALGORITMI Research Centre / LASI, Dept. of
Informatics, University of Minho, Braga,
Portugal

José Carlos L. Ramalho  (12)
University of Minho, Braga, Portugal

Eugénio Ribeiro  (3, 4, 7, 9)
INESC-ID Lisboa, Portugal; Instituto
Universitário de Lisboa (ISCTE-IUL), ISTAR,
Lisboa, Portugal

Leonardo Sampaio Rocha  (7)
Universidade Federal do Ceará, Fortaleza, Brasil

Christophe Scholliers  (20)
Ghent University, Belgium

Dietmar Seipel  (13)
University of Würzburg, Germany

Maarten Steevens  (20)
Ghent University, Belgium

António Teixeira  (16)
IEETA, DETI, University of Aveiro, Portugal;
LASI – Intelligent System Associate Laboratory,
Guimarães, Portugal

Alice Vieira  (10)
CLUNL – Research Center for Linguistics at
NOVA University Lisbon, NOVA University
Lisbon, Portugal

Ramon Vilanova Arbos  (14)
Dept. de Telecomunicació i Enginyeria de
Sistemes, Escola d'Enginyeria, Universitat
Autònoma de Barcelona, Spain

Xiaolan Xu  (8)
Faculty of Arts and Humanities, University of
Lisbon, Portugal

Inês Águia  (16)
IEETA, DETI, University of Aveiro, Portugal;
LASI – Intelligent System Associate Laboratory,
Guimarães, Portugal

■ Committees

Organization Committee

Alberto Simões
2Ai, School of Technology, IPCA, Portugal

André L. Santos
Iscte-IUL, ISTAR-IUL, Portugal

Eugénio Ribeiro
Iscte-IUL, ISTAR-IUL & INESC-ID Lisboa, Portugal

Fernando Batista
Iscte-IUL & INESC-ID Lisboa, Portugal

Ricardo Ribeiro
Iscte-IUL & INESC-ID Lisboa, Portugal

Diana Santos
ILOS, Universitetet i Oslo, Norway

Dietmar Seipel
University of Würzburg, Germany

Eugénio Ribeiro
Iscte-IUL, ISTAR-IUL & INESC-ID Lisboa, Portugal

Fernando Batista
Iscte-IUL & INESC-ID Lisboa, Portugal

Filipe Portela
Universidade do Minho, Portugal

Geylani Kardas
Ege University, Turkey

Irene Rodrigues
Universidade de Évora, Portugal

Ivan Lukovic
University of Belgrade, Serbia

Jakub Swacha
University of Szczecin, Poland

Jan Janousek
Czech Technical University, Czech Republic

Jaroslav Poruban
Technical University of Kosice, Slovakia

João Paulo Fernandes
New York University Abu Dhabi, United Arab Emirates

João Saraiva
Universidade do Minho, Portugal

Jorge Baptista
Universidade do Algarve & INESC-ID Lisboa, Portugal

José Barateiro
Universidade do Algarve, Portugal

José Carlos Paiva
Instituto Politécnico do Porto, Portugal

José João Almeida
Universidade do Minho, Portugal

Scientific Committee

Alberto Simões
2Ai, School of Technology, IPCA, Portugal

Alexander Paar
Duale Hochschule Sachsen, Germany

Ana Alves
CISUC & Instituto Politécnico de Coimbra, Portugal

Ana Rita Peixoto
Iscte-IUL, ISTAR-IUL, Portugal

André L. Santos
Iscte-IUL, ISTAR-IUL, Portugal

António Rosado da Cruz
Instituto Politécnico de Viana do Castelo, Portugal

António Teixeira
IEETA, DETI, LASI, Universidade de Aveiro, Portugal

Bostjan Slivnik
University of Ljubljana, Slovenia

Brett Drury
Liverpool Hope University, United Kingdom

Daniela da Cruz
DataDog, Portugal

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos



OpenAccess Series in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

José Luís Sierra
Universidad Complutense de Madrid, Spain

José Paulo Leal
Faculdade de Ciências da Universidade do
Porto, Portugal

Josep Silva
Universitat Politècnica de València, Spain

Marcos Garcia
CiTIUS, Universidade de Santiago de
Compostela, Spain

Maria João Varanda Pereira
Instituto Politécnico de Bragança, Portugal

Mario Beron
Universidad Nacional de San Luis, Argentina

Marjan Mernik
University of Maribor, Slovenia

Pablo Gamallo
CiTIUS, Universidade de Santiago de
Compostela, Spain

Pedro Rangel Henriques
Universidade do Minho, Portugal

Raquel Amaro
FCSH, Universidade NOVA de Lisboa,
Portugal

Ricardo Queirós
ESMAD, Instituto Politécnico do Porto &
INESC TEC, Portugal

Ricardo Ribeiro
Iscte-IUL & INESC-ID Lisboa, Portugal

Ricardo Rocha
Faculdade de Ciências da Universidade do
Porto, Portugal

Ricardo Rodrigues
CISUC & Instituto Politécnico de Coimbra,
Portugal

Ruben Solera Ureña
INESC-ID Lisboa, Portugal

Salvador Abreu
Universidade de Évora, Portugal

Sérgio Paulo
INESC-ID Lisboa, Portugal

Simão Melo de Sousa
Universidade do Algarve & NOVA LINCS,
Portugal

Optimising Retrieval for Linguistic Question-Answering in European Portuguese: A Benchmark on Ciberdúvidas Da Língua Portuguesa

Pedro Moura 

ISCTE – Instituto Universitário de Lisboa, Portugal

Inês Gama 

ISCTE – Instituto Universitário de Lisboa, Portugal

Fernando Batista 

ISCTE – Instituto Universitário de Lisboa, Portugal

INESC-ID Lisboa, Portugal

António Lopes 

ISCTE – Instituto Universitário de Lisboa, Portugal

ISTAR, Lisboa, Portugal

Abstract

Information retrieval for question-answering remains underexplored in specialised domains and under-resourced language variants such as European Portuguese. Existing benchmarks largely target general-domain English data and document-centric retrieval, failing to capture the semantic alignment required for linguistic consultation tasks over curated question-answer (QA) pairs. We address this gap by introducing a controlled evaluation framework for retrieval over the “Ciberdúvidas da Língua Portuguesa” corpus, comprising 29,145 expert-validated QA entries. Our approach systematically analyses the interaction between indexing strategies, encoder models, and retrieval paradigms, while modelling real-world query variability through a paraphrase-based benchmark of 600 queries across five user profiles, manually validated by a professional linguist to ensure semantic fidelity. Experiments show that dense retrieval with an IR-optimised monolingual encoder significantly outperforms both sparse (BM25) and hybrid methods, achieving a Mean Reciprocal Rank (MRR) of 0.93. Notably, hybrid retrieval underperforms due to lexical mismatch interference, challenging prevailing assumptions in the literature. Our contributions include a novel benchmark framework for linguistic QA retrieval, empirical evidence supporting monolingual IR-specialised models, and insights into retrieval robustness under paraphrastic variation, enabling improved QA systems for specialised and low-resource environments.

2012 ACM Subject Classification Information systems → Question answering; Information systems → Retrieval models and ranking; Information systems → Evaluation of retrieval results; Information systems → Digital libraries and archives

Keywords and phrases Information Retrieval, Question-Answering, European Portuguese, Sentence Encoders, Natural Language Processing

Digital Object Identifier 10.4230/OASICS.SLATE.2026.1

Funding This work was supported by Portuguese national funds through Fundação para a Ciência e a Tecnologia, I.P. (FCT) under projects UID/50021/2025 (DOI: <https://doi.org/10.54499/UID/50021/2025>), UID/PRR/50021/2025 (DOI: <https://doi.org/10.54499/UID/PRR/50021/2025>), and UID/04466/2025 (DOI: <https://doi.org/10.54499/UID/04466/2025>).



© Pedro Moura, Inês Gama, Fernando Batista, and António Lopes;
licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 1; pp. 1:1–1:17

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Information retrieval in specialised domains remains a central challenge in Natural Language Processing, particularly for languages that are under-represented in benchmark datasets and embedding model development. While retrieval systems have seen substantial progress in general-domain settings [17], their performance in certain fields and contexts remains underexplored. One domain where such systems can provide immediate value is language consultation. *Ciberdúvidas da Língua Portuguesa*¹ is a reference platform dedicated to clarifying doubts about the Portuguese language, covering topics such as spelling, phonetics, etymology, syntax, semantics, and pragmatics [7]. Over nearly three decades, the platform has accumulated more than 29,000 expert-curated responses, forming a rich and authoritative linguistic knowledge base. However, user interaction is currently limited to traditional keyword-based search, which constrains accessibility and makes it difficult to provide immediate, context-aware answers to recurring queries.

Deploying an effective natural language retrieval system over such a corpus presents distinct challenges. First, query formulation varies widely across users, where the same linguistic doubt may be expressed synthetically, formally, informally, or using domain-specific pedagogical terminology. In addition, the corpus consists of questions and answers in Portuguese, mostly in its European variant, a variety that has received comparatively little attention in the development of models [14]. Existing retrieval benchmarks predominantly target general-domain, well-formed queries in high-resource languages, offering limited guidance for specialised linguistic corpora in under-represented language variants. A controlled evaluation of how indexing strategies, embedding models, and retrieval paradigms interact in this setting is also notably absent from the literature.

To address this gap, we propose an evaluation framework for document retrieval that explicitly models real-world query variability through structured paraphrasing profiles. This work focuses on the design, systematic benchmarking, and evaluation of retrieval strategies over the “Ciberdúvidas da Língua Portuguesa” corpus. Rather than assuming a single optimal retrieval configuration, we conduct a controlled experimental analysis of how different preprocessing and indexing strategies, embedding models and retrieval paradigms affect the alignment between user queries and expert linguistic knowledge.

2 Related Work

The development of an effective retrieval system requires a comprehensive understanding of current paradigms in Information Retrieval. This section outlines the foundational literature and recent advancements pertinent to our study.

2.1 Retrieval Targets in Question-Answering Environments

In retrieval applied to question-answering environments, the literature considers different approaches to system design.

The first paradigm treats the corpus as generic passages and attempts to identify an answer to the user’s query. This is predominantly seen in open-domain question-answering, where the goal is to search a large collection of documents to retrieve a small subset of candidate passages that might contain the factual answer [13], as presented in DPR. These architectures

¹ Ciberdúvidas da Língua Portuguesa. Available at: <https://ciberduvidas.iscte-iul.pt>

utilize a dual-encoder framework to map both the question and the text passages into a low-dimensional space, enabling the retriever to capture semantic relationships even when synonyms or paraphrases consist of completely different tokens [13]. In open-domain question-answering, these systems are commonly modelled under the retriever–reader paradigm, where a retrieval component first selects relevant documents and a reader subsequently extracts the final answer [28].

In contrast, question–answer retrieval systems typically do not require an explicit reader component, as they operate by ranking pre-existing QA pairs rather than extracting answers from raw text. A major challenge in this environment is the lexical gap which occurs in contexts where users’ natural language queries are short and lack overlapping vocabulary with the pre-recorded FAQ pairs [12]. In [12], researchers have modeled the task as a relevance classification problem, employing a variety of semantic textual similarity features to match user queries with the corresponding FAQ. The research [5], applied to a FAQ corpus in European Portuguese, follows this line of development, directly calculating similarity scores between the user’s query vector and the vectors of the queries in the corpus, and selecting the FAQ with the highest similarity.

Hybrid approaches such as the Multi-Field Bi-Encoder (MFBE) [3] exploit the multi-field structure of QA pairs, combining title, question body, and category, demonstrating that multi-field representation significantly outperforms approaches based solely on the title. Similarly, [16] proposes an initial re-ranking based on lexical similarity, followed by two BERT models dedicated respectively to query–question and query–answer matching, combining all re-rankers via late fusion.

2.2 Encoding Models and the Portuguese Language

The selection of the embedding model is critical to the performance of the retrieval system [8, 9], as it dictates how effectively the semantic meaning of user queries is mapped to the relevant documents in the knowledge base. The scope of this research is not to identify the absolute best model for this purpose; the experiments conducted are designed as a proof of concept. It will therefore be essential to examine the state-of-the-art (SOTA) for Portuguese language embeddings, as well as a multilingual baseline alternative based on SBERT. In [18], a performance comparison between monolingual and multilingual encoding models for Italian demonstrates that monolingual models exhibit greater sensitivity to language-specific syntactic and semantic patterns. Similar findings have been reported for Portuguese embedding models. In [10], for instance, Serafim outperformed all multilingual models included in the comparison. These results highlight the importance of exploring embedding models specifically designed for Portuguese and its linguistic variants.

The formalization of models focused on European Portuguese is an area that has been largely unexplored in the literature [14]. In the literature consulted, no surveys or comparative studies were found among the most recent models developed. Despite this, one main model with the potential to contribute to this research was noted. The Albertina family [20], provides open-source encoder foundation models for European and Brazilian Portuguese. Two models were created, with 100M parameters and 1.5B parameters, based on the DeBERTa architecture. These models were evaluated and obtained good results in STS (Semantic Textual Similarity) tasks, thus indicating high performance in embedding generation.

The Serafim family represents a set of open-source sentence encoders specifically developed for the Portuguese language, covering both the European (PT-PT) and Brazilian (PT-BR) variants [10]. Built upon foundation models such as Albertina and BERTimbau, Serafim establishes a new state of the art [10] for Portuguese in both Semantic Textual Similarity

(STS) and Information Retrieval (IR) benchmarks, significantly outperforming previous Portuguese-specific encoders and multilingual baselines. The 900M parameter version, based on Albertina PT-PT, demonstrated superior performance specifically on European Portuguese datasets [10]. The Serafim 900M model is available in two variants: a general-purpose sentence embedding model and an Information Retrieval (IR)-optimized version. For clarity and consistency throughout this article, the model identified as `serafim-900m-portuguese-pt-sentence-encoder` will be referred to as **Serafim900m**, while the model `serafim-900m-portuguese-pt-sentence-encoder-ir` will be referred to as **Serafim900m-IR**.

2.3 Retrieval Benchmarks

The development of robust evaluation frameworks has been central to progress in information retrieval and question-answering research [2].

The MS MARCO dataset [2], constructed from over one million anonymized queries from Bing search logs, has been widely used in the field. This dataset has even been used to train some state-of-the-art encoders, such as the MiniLM and E5 families [25, 26]. Although this dataset is in English, the mMarco version [4] performed machine translation of this dataset into 13 languages, one of them Portuguese (no variant specified). To address the limitations of evaluating information retrieval in narrow settings, the Benchmarking-IR (BEIR) benchmark was introduced [23], consisting of a selection of various documents from a range of domains, including question-answering. This benchmark was developed using datasets in English exclusively [23].

Multilingual efforts have been made, for example, in the MIRACL [27] dataset, which supports 18 languages (though Portuguese is not included). This dataset was specifically designed by human annotators to enable the evaluation of retrieval tasks in which the natural-language query and the corpus are in the same language.

Existing benchmarks are primarily designed for general-purpose information retrieval, where queries correspond to factual information needs and documents are treated as independent evidence sources [23, 2, 4]. In contrast, linguistic consultation tasks exhibit different properties: queries are often metalinguistic, involving questions about grammar, orthography, or semantic nuance, where relevance depends on alignment between a user question and a previously formulated expert explanation. Furthermore, existing benchmarks typically assume a document-centric retrieval paradigm [27, 2, 4], whereas in linguistic QA settings the corpus is composed of curated QA pairs, introducing a matching problem closer to question similarity and intent alignment than to traditional passage retrieval.

These limitations motivate the need for a dedicated dataset tailored to the specific characteristics of linguistic question-answering in European Portuguese.

3 Ciberdúvidas Dataset

The corpus was derived from the internal database of *Ciberdúvidas* platform, comprising consultations published between 1997 and January 2026. Due to the nature of our institutional partnership with the platform, this corpus and the generated test dataset are not publicly available. As the original content is proprietary to the platform, the dataset is subject to their internal data policies and licensing.

After extraction and preprocessing, the resulting corpus contains exactly 29,145 consultation entries. Each entry contains a unique ID, a title summarizing the subject, the user’s question body (HTML), the expert’s answer (HTML), and a timestamp.

The corpus exhibits significant variability in both question formulation and response length. Questions range from short lexical clarifications to longer contextualized inquiries about grammatical structures or language usage. Expert responses similarly vary in length, often including detailed explanations, historical context, and illustrative examples.

To better understand the linguistic phenomena present in the *Ciberdúvidas* collection, a latent topic analysis based on BERTopic was performed on the corpus. The distribution of queries reveals a predominant interest in etymology and word origins, as well as frequent concern with how particular sentences should be written. Furthermore, the corpus comprises a substantial volume of inquiries pertaining to phraseological semantics, idiomatic expressions, and syntactic analysis.

While the consultation service primarily produces content in European Portuguese, the dataset inevitably includes questions originating from other Portuguese variants. Users from diverse Portuguese-speaking regions may submit inquiries that exhibit lexical, orthographic, or syntactic patterns characteristic of Brazilian, African, or other Portuguese varieties. These instances are not explicitly annotated in the metadata, introducing a degree of linguistic variability in the corpus.

Considering token counts, the questions have an average length of 100.9 ± 124.2 tokens, ranging from 7 to 5542 tokens, while answers have an average length of 331.6 ± 319.3 tokens, ranging from 6 to 5756 tokens (all token counts were computed using the *Serafim900m-IR* model, after a basic HTML tag removal preprocessing).

4 Experimental Setup

To evaluate and choose the retrieval techniques to be used, a testing framework was developed to assess the robustness of different models, preprocessing methods, and approaches.

Figure 1 illustrates the complete experimental framework used in this study, which is suitable for generalization and application in other domains.

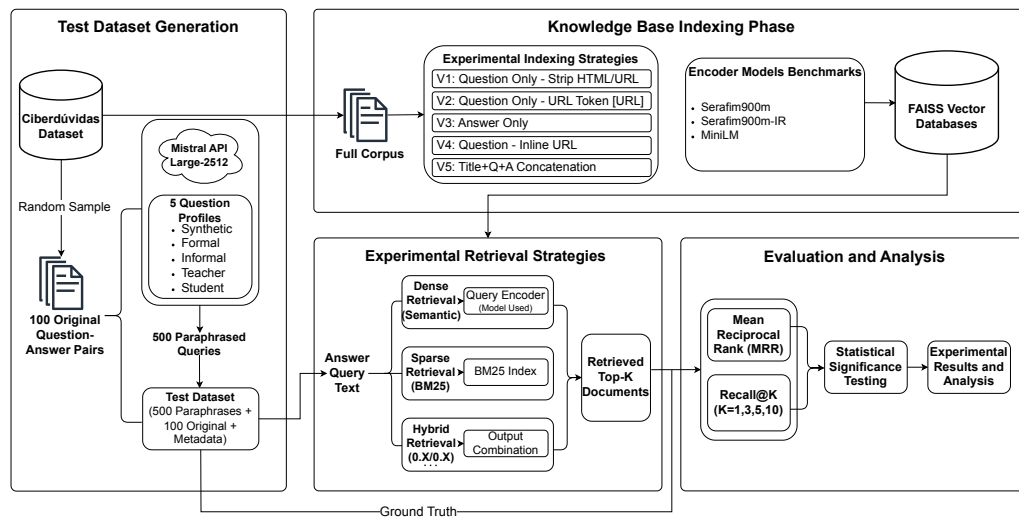


Figure 1 Overview of the experimental framework, illustrating the test dataset generation, knowledge base indexing phase, experimental retrieval strategies, and the evaluation pipeline.

4.1 Test Dataset Generation and Paraphrasing

In this context, it will be essential to correctly match each user’s query with the appropriate document, regardless of the user’s articulation, formality, or vocabulary, bridging “the gap between query and document vocabulary” [17]. To simulate real-world variance, an evaluation dataset was constructed by sampling 100 original questions from the Ciberdúvidas corpus, testing the retriever’s ability to find documents related to the same query but phrased differently. This paraphrase-based testing approach was used in a similar context by [5], albeit on a small set of manually constructed questions.

Inspired by this approach, to generate the paraphrases, the “mistral-large-2512” model was used through the Mistral API. This model was chosen due to its native compatibility with the Portuguese language, performance, and capacity for structured outputs [1].

The generation of paraphrases was based on the creation of five different query profiles, which aim to portray different question approaches from the target audience of Ciberdúvidas:

- **Synthetic:** Short, direct, search-engine style queries.
- **Formal/Explicit:** Grammatically rigorous and highly detailed queries.
- **Informal/Natural:** Casual, conversational tone queries.
- **Portuguese Language Teacher:** Queries utilising technical pedagogical and linguistic terminology.
- **High School Student:** Direct, comprehension-focused queries.

This methodology yielded a final test suite of 600 queries (100 original + 500 paraphrased variations). Table 1 illustrates a representative example of how a single original consultation was transformed across the five distinct profiles.

■ **Table 1** Example of paraphrases generated for a specific question.

Profile	Portuguese Query	English Translation
Original	Qual é o gentílico de Costa do Marfim?	<i>What is the demonym of Ivory Coast?</i>
Synthetic	Gentílico de Costa do Marfim?	<i>Demonym of Ivory Coast?</i>
Formal	Poderia indicar-me, com precisão, qual é o termo que designa os naturais ou habitantes da Costa do Marfim?	<i>Could you accurately indicate the term that designates the natives or inhabitants of Ivory Coast?</i>
Informal	Então, como é que se chama uma pessoa que é da Costa do Marfim? Não me lembro bem...	<i>So, what do you call a person who is from Ivory Coast? I can’t quite remember...</i>
Teacher	No âmbito da onomástica geográfica, qual é o etnónimo correspondente ao topónimo ‘Costa do Marfim’, utilizado para denominar os seus cidadãos ou naturais?	<i>In the context of geographical onomastics, what is the ethnonym corresponding to the toponym ‘Ivory Coast’, used to denominate its citizens or natives?</i>
Student	Na aula falámos dos nomes das pessoas de cada país, mas não me recordo qual é o nome certo para quem nasce na Costa do Marfim. Pode ajudar?	<i>In class we talked about the names of people from each country, but I don’t remember the right name for someone born in Ivory Coast. Can you help?</i>

4.2 Expert Validation of Dataset

To ensure the relevance and quality of the generated paraphrases, a manual validation was conducted. A specialised linguist from the Ciberdúvidas team performed a blind review of the entire evaluation dataset, comprising all 600 queries.

In this process, the evaluator was provided with the target expert answer alongside the corresponding six query variations (the original query and the five generated paraphrases), but was kept blind to the specific tags assigned to each generated variation. The main objective of this review was to assess whether the synthetic queries maintained semantic fidelity to the original problem without introducing misrepresentations. The linguist categorised each query into one of three classifications:

- **Valid:** The query accurately reflects the original intent and makes sense contextually and linguistically.
- **Valid but with adjusted scope:** The query maintains the core semantic value but introduces a shift in focus, such as deepening the theoretical premise, demanding a more elaborate explanation or using overly specific constraints compared to the original question.
- **Invalid:** The query fails to capture the original intent.

The outcome of the evaluation demonstrated high efficacy of the framework. Out of the 600 queries analysed, 573 (95.5%) were classified as Valid. A further 22 queries (3.6%) were classified as Valid but with adjusted scope. This occurred primarily in the “Teacher” profile, where there was a tendency to expand the query by including morphologically complex vocabulary and requesting additional details. Only 5 queries (0.8%) were deemed Invalid, underscoring the reliability of the dataset. In addition to the categorization, the expert provided positive qualitative feedback, noting that the dataset is highly relevant and presents a realistic challenge for the Ciberdúvidas, mirroring the diverse real-world formulations encountered by the Ciberdúvidas platform on a daily basis. Despite these minor deviations, all 600 queries were retained in the final benchmark to faithfully represent the inherent noise and edge cases present in real-world user submissions and automated query processing.

4.3 Data Indexing Strategies

To evaluate the impact of data representation on retrieval performance, several indexing strategies were explored by varying two key dimensions: the target text selected for embedding and the preprocessing applied to the textual content. Rather than present all possible configurations, we selected a subset of strategies that we considered most representative and relevant for the study.

Regarding the object of encoding, three main targets were explored: the Question, representing the user’s original information need and serving as the primary retrieval anchor; the Answer, which emphasises the resolution or explanation and enables a query-to-answer matching paradigm; and a combined Title + Question + Answer, providing a more holistic representation by aggregating multiple semantic signals, albeit at the cost of increased verbosity.

In parallel, different preprocessing strategies were applied to assess their impact on embedding quality. Across all variants, a baseline normalization pipeline ensured textual consistency, including the conversion of typographical quotes, the collapsing of multiple whitespace characters, and the trimming of leading and trailing spaces. Beyond this baseline, variations focused on the treatment of URL elements: URLs could be completely removed from the text and stored in metadata, replaced with a placeholder token (e.g., [URL]) while preserving original links in metadata, or kept inline in a human-readable format (e.g., `text (url)`).

The combinations of these elements have given rise to various variants, described below:

- **V1** – Target is the Question. HTML and URLs are stripped and moved to metadata. *Primary Objective:* Baseline semantic matching.
- **V2** – Target is the Question. URLs are replaced with a [URL] token, and original links are moved to metadata. *Primary Objective:* Preserve link context without noise.
- **V3** – Target is the Answer. HTML and URLs are stripped and moved to metadata. *Primary Objective:* Shift semantic focus to resolution.
- **V4** – Target is the Question. HTML is stripped, but URLs are kept inline formatted as `text (url)`. *Primary Objective:* Retain explicit web references.
- **V5** – Target is the Title + Question + Answer. HTML and URLs are stripped and moved to metadata. *Primary Objective:* Holistic semantic representation.

4.4 Encoder Models

To evaluate the impact of embedding representations on retrieval accuracy, we benchmarked a diverse set of three encoder models. These models were selected to compare the efficacy of European Portuguese-specific training against state-of-the-art multilingual architectures, as well as to assess the impact of model size and task-specific fine-tuning. The evaluated models include:

- **Serafim Family (900M):** As introduced previously, these represent the state-of-the-art for the Portuguese language. We evaluated both the base sentence encoder (**Serafim900m**) and its Information Retrieval-optimised variant (**Serafim900m-IR**) to isolate and measure the impact of IR-specific loss functions.
- **Paraphrase-Multilingual-MiniLM-L12-v2:** Selected as a lightweight baseline to determine the performance floor of a smaller model. For clarity and consistency throughout this article, this model will be referred to as MiniLM.

4.5 Retrieval Strategies

To benchmark performance, three distinct retrieval strategies were evaluated for the top $k=10$ documents:

Dense Retrieval (Semantic FAISS): This strategy utilises the high-dimensional embeddings generated by the models to perform semantic similarity searches. By mapping queries and documents into a shared vector space, it identifies relevant content based on linguistic meaning rather than exact word matches.

Sparse Retrieval (BM25): A traditional lexical matching algorithm that serves as a statistical baseline. utilising the **BM25Retriever**, this approach ranks documents based on term frequency and inverse document frequency, effectively capturing exact keyword overlaps and technical terminology.

Hybrid Retrieval: This approach employs an **EnsembleRetriever** that combines the outputs of both BM25 and Semantic FAISS via Reciprocal Rank Fusion, where different weight combinations were empirically tested by Grid search (e.g., 0.1/0.9, 0.2/0.8, ..., 0.9/0.1) to assess the relative contribution of the semantic (Dense Retrieval) and lexical (BM25) components, enabling the selection of the configuration that maximizes overall system performance.

4.6 Evaluation Metrics

Given the current context, we are dealing with a Single-Target Retrieval scenario, since for each query, there is (theoretically) only one correct document. Based on this premise, the system’s main objective is to place the correct document as high as possible in the ranking. To evaluate this, we selected the following metrics:

- **Mean Reciprocal Rank (MRR):** Serves as the primary metric for this study. It is expressed as:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$

where rank_i is the position of the first relevant document for the i -th query [24].

- **Recall@K ($R@k$):** This metric allows us to observe a Hit Rate scenario. It is expressed by:

$$R@k = \frac{|\text{Relevant Documents} \cap \text{Retrieved Documents}@k|}{|\text{Total Relevant Documents}|}$$

If the document is in the top K, Recall is 1; otherwise, it is 0. This directly translates to the probability that the answer is among the first K results [11, 24, 6].

Thus, in light of the literature, we chose to exclude other prominent metrics, such as Precision@K, as we consider that these will not add essential information to the case.

To assess whether observed differences in retrieval performance are statistically significant, we adopt paired significance tests commonly used in Information Retrieval evaluation. Following the recommendations of [21], we employ Fisher’s randomization test (permutation test), as it directly evaluates differences in the metric of interest without transforming scores into ranks. This approach avoids the loss of magnitude information associated with rank-based tests such as the Wilcoxon signed-rank test. Although the study [21] focused on the Mean Absolute Precision metric, the researchers note similar behaviour when applied to the Mean Reciprocal Rank.

All tests in these experiments are conducted over the full set of 600 queries, over Mean Reciprocal Rank, using paired observations per query. The permutation test was computed using 100,000 random shuffles and significance is determined at the $\alpha = 0.05$ level.

5 Experiments

To systematically evaluate the proposed architecture, we conducted a comprehensive series of experiments analyzing three primary dimensions: the impact of the indexing strategy, the performance of the encoder models, and the efficacy of different retrieval paradigms.

5.1 Indexing Strategy Study

The first phase of our evaluation aimed to determine the optimal textual representation for the vector database. To isolate this variable, we established a control environment utilising the IR-specialised encoder (Serafim900m-IR) and the dense retrieval method (Semantic FAISS). We evaluated the five preprocessing variants (V1 through V5).

To rigorously evaluate the marginal differences between V1 and V2, a Randomization test (Fisher’s permutation test) was conducted across the 600 paired queries for comparison. The randomization test revealed no statistically significant difference in performance ($p = 0.12240$),

■ **Table 2** Indexing Strategies Results.

Variant	MRR	R@1	R@3	R@5	R@10
V1	0.9340	0.9200	0.9450	0.9533	0.9567
V2	0.9320	0.9167	0.9467	0.9533	0.9550
V3	0.6217	0.5550	0.6767	0.7083	0.7383
V4	0.9276	0.9083	0.9450	0.9517	0.9550
V5	0.9314	0.9083	0.9500	0.9617	0.9700

confirming that the inclusion of the [URL] token does not mathematically degrade semantic retrieval. These results suggest that URL treatment has no substantial effect on semantic retrieval performance. This shows the model’s ability to handle these nuances in the text. With regard to V3, it is interesting to note results that clearly differ from those observed in [19], since in this context this strategy proved to be of little use in all metrics. This may have been partly due to the truncation of documents, as the context was exceeded in 5,221 cases. The V1 variants had the highest MRR and Recall@1, with a very small difference to the V5 approach. A statistical comparison between these approaches also revealed no significant difference ($p = 0.93605$), leading to the conclusion that there is no statistical evidence to distinguish between the two approaches.

Given this, it is interesting to look at other metrics that allow for an objective conclusion about the strategy to choose.

■ **Table 3** Token count statistics across the five indexing variants. All variants consist of exactly 29,145 documents. Tokens were calculated using the `Serafim900m-IR` tokenizer.

Variant	Average Tokens ($\mu \pm \sigma$)	Min Tokens	Max Tokens	>512 Tokens
V1	98.1 $\pm$ 123.7	5	5540	361
V2	98.1 $\pm$ 123.5	5	5540	360
V3	326.9 $\pm$ 316.6	4	5743	5221
V4	101.4 $\pm$ 128.7	5	5540	400
V5	446.7 $\pm$ 374.3	32	7839	8439

Analysis of token count statistics across the indexing variants shows that V1, V2, and V4 maintain relatively low average token lengths (~ 100 tokens), while V3 and V5 significantly increase the number of tokens due to the inclusion of expert answers and full document context.

Given the `Serafim900m-IR` model’s limit of 512 tokens, we also concluded that some documents suffered from truncation, with part of the response being truncated (although stored in metadata for future context). This result is even more significant in V5, with 8439 documents undergoing this process.

Based on the limitations described, V2 will be evaluated in subsequent phases. Although the quantitative advantage does not show clear improvements over approach 1 and 4, the presence of a placeholder indicating a URL helps the model to syntactically structure the sentence, ensuring there is no abrupt break between words. In addition, this variant demonstrates excellent token efficiency when compared to V5.

5.2 Encoder Model Benchmark

Having established the V2 indexing strategy as the optimal baseline for semantic alignment, the second phase of our evaluation focuses on benchmarking the performance of different text-encoder models. To strictly isolate the embedding quality and semantic representation capabilities of each model, this comparative analysis evaluates performance exclusively under the dense retrieval paradigm (Semantic FAISS), omitting sparse lexical features.

■ **Table 4** Global dense retrieval performance (Semantic FAISS) of encoder models using the V2 indexing strategy.

Encoder Model	MRR	R@1	R@3	R@5	R@10
Serafim900m	0.8705	0.8417	0.8900	0.9100	0.9267
Serafim900m-IR	0.9320	0.9167	0.9467	0.9533	0.9550
MiniLM	0.7052	0.6767	0.7217	0.7400	0.7717

As shown in Table 4, the Serafim900m-IR model delivered the best performance across all metrics, achieving an MRR of 0.9320. It is nevertheless important to note that the results obtained by MiniLM provide interesting data for a multilingual baseline, making it an acceptable lightweight alternative.

To ensure that the performance gap between the base Serafim900m model and its Information Retrieval variant was not due to statistical noise, another statistical test was conducted across the 600 paired queries. The test confirmed that the superiority of Serafim900m-IR over Serafim900m is statistically significant ($p = 0.00002$), validating the benefit of task-specific fine-tuning.

5.3 Retrieval Strategy Comparison

For the retrieval strategy analysis, we fixed the text representation to Variant 2 and analysed the performance using the top-performing model (Serafim900m-IR).

For this analysis, not only were simple retrieval strategies (Semantic and BM25) used, but a hybrid approach employing Reciprocal Rank Fusion was also adopted. To parametrise this approach, a grid search was carried out, as shown in Table 5.

■ **Table 5** Results for Retrieval Strategies with (Semantic / BM25).

Retriever (Semantic / BM25)	MRR	R@1	R@3	R@5	R@10
0.0 / 1.0 (BM25)	0.6455	0.6267	0.6583	0.6683	0.6900
1.0 / 0.0 (Semantic)	0.9320	0.9167	0.9467	0.9533	0.9550
0.1 / 0.9	0.6957	0.6550	0.6883	0.6883	0.6933
0.2 / 0.8	0.7015	0.6650	0.6883	0.6883	0.6933
0.3 / 0.7	0.7015	0.6650	0.6883	0.6883	0.6933
0.4 / 0.6	0.7015	0.6650	0.6883	0.6883	0.6933
0.5 / 0.5	0.8073	0.6800	0.9283	0.9533	0.9650
0.6 / 0.4	0.9042	0.8633	0.9450	0.9550	0.9550
0.7 / 0.3	0.9061	0.8667	0.9450	0.9550	0.9550
0.8 / 0.2	0.9061	0.8667	0.9450	0.9550	0.9550
0.9 / 0.1	0.9124	0.8767	0.9467	0.9550	0.9550

The results, presented in Table 5, reveal a clear directional trend: retrieval performance scales almost monotonically with the semantic weight.

Configurations heavily reliant on lexical matching (semantic weights from 0.1 to 0.4) performed poorly, plateauing around a Mean Reciprocal Rank (MRR) of 0.7015. As the semantic weight increased past the 0.5 threshold, a significant performance leap occurred, jumping to an MRR of 0.9042 at the 0.6/0.4 split. The best hybrid performance was achieved with the maximum semantic weighting, in the 0.9/0.1 strategy, with an MRR of 0.9124.

As shown, the MRR of dense retrieval strategy significantly outperforms both the Sparse (BM25) and the best Hybrid approach across all metrics. BM25 establishes a baseline MRR of 0.6455, suffering visibly from the “vocabulary mismatch” problem when users formulate queries using terminology distinct from the indexed documents.

In contrast to prevailing trends in the literature, where hybrid retrieval with dense and sparse methods, is generally assumed to “capture different relevance features and (...) benefit from each other by leveraging complementary relevance information” [8], our ensemble approach consistently underperformed the purely semantic strategy. The BM25 component frequently ranked lexically similar but semantically misaligned documents highly. When combined through weighted rank fusion, these noisy lexical rankings displaced correct dense-retrieval results, reducing the MRR.

It is also worth noting the interesting result obtained in R@10 with the (0.5/0.5) configuration. This showed a slight improvement over semantic retrieval, suggesting that further development of a reranker could potentially lead to an increase in MRR in these hybrid strategies.

6 Analysis and Discussion

To fully grasp the operational realities of deploying this architecture, it is necessary to examine how our chosen configuration interacts with the inherent semantic and structural complexities of natural user queries. Therefore, it is essential to explore the system’s robustness across diverse user profiles, investigate specific linguistic patterns that influence retrieval difficulty, and critically assess the limitations of our current framework to contextualize the broader contributions and future directions of this research.

6.1 Robustness Across Query Profiles

To better understand the retrieval limitations, we disaggregated the performance metrics across the five synthesized query profiles. This analysis allows us to observe how different user articulations impact the system’s ability to locate the correct document.

■ **Table 6** Mean Reciprocal Rank (MRR) by Query Profile using Variant 2.

Query Type	Sparse	Semantic FAISS	
	BM25	Serafim900m-IR	miniLM
Original	1.0000	1.0000	1.0000
Formal	0.5654	0.9045	0.6230
Informal	0.6215	0.9575	0.8215
Synthetic	0.7041	0.9583	0.7482
Student	0.5636	0.9417	0.6934
Teacher	0.4185	0.8299	0.3449

It is interesting to note, first of all, that the differences between profiles in BM25 are much more pronounced than those obtained using semantic retrieval. This demonstrates that this type of retrieval is highly adaptable to vocabulary mismatches.

In addition, the query type that consistently yields the worst results is the “teacher” query. This can be seen in the analysis of an example query from the test set:

Original Question: Qual é a diferença entre vem, vêm e veem? [Eng: What is the difference between “vem” (coming), “vêm” (are coming) and “veem” (see)?]

Teacher: No âmbito do paradigma flexional do verbo vir e do verbo ver no presente do indicativo, como se articulam as formas “vem”, “vêm” e “veem” em termos de número, pessoa gramatical e valor semântico, considerando as especificidades do português europeu? [Eng: Within the inflectional paradigm of the verbs “vir” (come) and “ver” (see) in the present indicative, how do the forms “vem” (coming), “vêm” (are coming), and “veem” (see) relate in terms of number, grammatical person, and semantic value, considering the specificities of European Portuguese?]

It is clear from the example above that, in some cases, the model not only expanded the vocabulary of the original question with specific technical terminology but also gave the query a more specific purpose, as previously highlighted by the linguistic expert. This is not a problem, since, in future real-world applications, it is desirable for questions (even those with different objectives but similar scopes), to serve as context for the user or system using this retriever. Nevertheless, this makes this paraphrase profile more challenging and explains why the results obtained, although high, are lower than those observed in other paraphrase profiles.

6.2 Query Difficulty and Patterns Observed

The results across the various query profiles show that query difficulty differs markedly from one profile to another. Consequently, it is crucial to analyze specific examples of the generated queries and determine the features that characterize the paraphrasing for each profile.

One of the challenges identified involves queries that focus on a direct quote, a specific word, or other elements that require reference throughout the paraphrases, without the possibility of using synonyms. For example:

Original Question: No exemplo «Eu fui lá, à praia», estamos perante um complemento oblíquo («lá, à praia»), ou dois complementos ligados por uma vírgula? [Eng: In the example “I went there, to the beach,” are we dealing with an oblique complement (“there, to the beach”), or two complements connected by a comma?]

Informal: Olha, na frase «Eu fui lá, à praia», o «lá, à praia» é um complemento só ou são dois? Tipo, a vírgula separa mesmo duas coisas diferentes? – [Eng: Look, in the sentence “I went there, to the beach,” is “there, to the beach” a single complement or are there two? Like, does the comma really separate two different things?...]

In this exact case, the question exhibited a 100% retrieval success rate. The inclusion of direct, word-for-word quotations certainly contributed to this, since the paraphrases, while adapting the context to the profile, had to retain a quotation to give the question its purpose.

The same holds in some examples that include large quoted passages from books or similar sources. This creates a dilemma: Leaving such examples in the test set could indirectly inflate performance metrics; on the other hand, they reflect real user questions that are likely to reappear, and their successful retrieval indicates that the system is functioning

correctly. Moreover, removing these questions from the test set would require a clear, specific rule, which inevitably introduces subjectivity: Should all questions containing quotations be removed from the test set? Only those with quotes longer than X words? On what basis?

6.3 Limitations and Future Work

The main limitation identified stems from the overestimation of the results obtained, particularly when examining the MRR metric. Given the nature of the proposed scenario, it was necessary to find the perfect balance between paraphrasing in a challenging way and altering the question so substantially that it no longer made sense for the retrieval evaluation. Consequently, in examples that relied on quotations or questions about exact words, these had to be maintained across the variations, substantially increasing the ease of retrieval. In addition, due to the syntactic nature of the dataset, it may not fully replicate certain characteristics of the end user, such as spelling errors.

Based on this analysis, an improved approach might involve creating a manual dataset based on an analysis of user interaction with the platform. This data could stem from interaction with a potential future conversational agent or natural language-based search system. These tools do not yet exist on the portal, but their future data could contribute to this development.

The observed performance gaps across query profiles and the underperformance of hybrid retrieval further motivate the exploration of advanced retrieval enhancement techniques. In the consulted literature, promising directions for future work emerge. First, the integration of reranking models [22] represents a natural extension of the current pipeline. In this approach, an initial set of candidate documents retrieved by the dense model would be re-evaluated and reordered using a cross-encoder or specialised reranker, potentially improving the precision of top-ranked results. Given that the current system already achieves high recall, reranking may be particularly effective in optimising ranking quality, especially in challenging or ambiguous queries. Also, query rewriting techniques [15, 6] could be explored to address the variability observed across query profiles, especially in more complex formulations such as the “Teacher” queries. By transforming user inputs into more standardized formulations, rewriting has the potential to further improve alignment between queries and indexed documents.

Given the solid foundation provided by the findings of this analysis, future work should also focus on evaluating solutions that enable the integration of this process into the platform. One of the key approaches to be evaluated involves the development of a Retrieval-Augmented Generation system, based on a similar process for searching and using these documents.

6.4 Contributions

First, this research provides a systematic analysis of the retrieval layer as the primary focus of study. Based on controlled experiments, we demonstrate how variations in indexing strategies, embedding models, and retrieval paradigms directly affect the system’s ability to align queries with the knowledge base. Furthermore, this study proposes an evaluation framework that can be extended to other domains of question-answering systems. By constructing a test dataset based on paraphrases, this work establishes a realistic benchmark for retrieval evaluation.

The inclusion of specialised validation reinforces this approach and ensures the viability of the strategy employed. Also, this study enables a direct comparison of a state-of-the-art monolingual model for Portuguese with a robust multilingual baseline, revealing clear improvements that are supported by the relevant literature. At the same time, the study highlights the significant gains obtained through task-specific fine-tuning, as demonstrated by the performance improvements of the IR-optimised Serafim model.

From an artefact-oriented perspective, the results of this study enable an immediate improvement to the retrieval system in the “*Ciberdúvidas da Língua Portuguesa*” project, when compared to the current keyword-based search mechanism, enabling the future work discussed earlier. Consequently, these contributions establish a solid foundation for future advancements, especially in the direction of integrating more sophisticated retrieval and generation techniques, ultimately supporting the development of more accurate and user-aligned question-answering systems.

References

- 1 Mistral AI. Mistral large 3 - mistral ai | mistral docs. URL: <https://docs.mistral.ai/models/mistral-large-3-25-12>.
- 2 Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. Ms marco: A human generated machine reading comprehension dataset. *CEUR Workshop Proceedings*, 1773, November 2016. [arXiv:1611.09268](https://arxiv.org/abs/1611.09268).
- 3 Debopriyo Banerjee, Mausam Jain, and Ashish Kulkarni. Mfbc: Leveraging multi-field information of faqs for efficient dense retrieval. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13937 LNCS:112–124, March 2023. doi:10.1007/978-3-031-33380-4_9.
- 4 Luiz Bonifacio, Vitor Jeronimo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Zeta Alpha, Roberto Lotufo, and Rodrigo Nogueira. mmarco: A multilingual version of the ms marco passage ranking dataset. *arXiv preprint arXiv:2108.13897*, August 2021. [arXiv:2108.13897](https://arxiv.org/abs/2108.13897).
- 5 Nuno Carriço and Paulo Quaresma. Sentence embeddings and sentence similarity for portuguese faqs. In *Proceedings of IberSPEECH 2021*, 2021. doi:10.21437/IberSPEECH.2021-43.
- 6 Daksh Chaudhary, Sri Lakshmi Vadlamani, Dimple Thomas, Shiva Nejati, and Mehrdad Sabetzadeh. Developing a llama-based chatbot for ci/cd question answering: A case study at ericsson. *Proceedings - 2024 IEEE International Conference on Software Maintenance and Evolution, ICSME 2024*, pages 707–718, 2024. doi:10.1109/ICSME58944.2024.00075.
- 7 Ciberdúvidas da Língua Portuguesa. Quem somos. URL: <https://ciberduvidas.iscte-iul.pt/quem-somos>.
- 8 Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *Proceedings - 2024 Conference on AI, Science, Engineering, and Technology, AIxSET 2024*, pages 166–169, December 2023. doi:10.1109/AIxSET62544.2024.00030.
- 9 Samira Ghodrathnama and Mehrdad Zakershahrak. Adapting llms for efficient, personalized information retrieval: Methods and implications. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 14518 LNCS:17–26, 2024. doi:10.1007/978-981-97-0989-2_2.
- 10 Luís Gomes, António Branco, João Silva, João Rodrigues, and Rodrigo Santos. Open sentence embeddings for portuguese with the serafim pt* encoders family. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 14969 LNAI:267–279, 2025. doi:10.1007/978-3-031-73503-5_22.
- 11 Md Saleh Ibtasham, Sarmad Bashir, Muhammad Abbas, Zulqarnain Haider, Mehrdad Saadatmand, and Antonio Cicchetti. Reqrq: Enhancing software release management through retrieval-augmented llms: An industrial study. *Lecture Notes in Computer Science*, 15588 LNCS:277–292, 2025. doi:10.1007/978-3-031-88531-0_20.
- 12 Vanja Mladen Karan, Lovro Žmak, and Jan Šnajder. Frequently asked questions retrieval for croatian based on semantic textual similarity. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 24–33. Association for Computational Linguistics, 2013. URL: <https://aclanthology.org/W13-2405/>.

- 13 Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen Tau Yih. Dense passage retrieval for open-domain question answering. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 6769–6781, 2020. doi:10.18653/V1/2020.EMNLP-MAIN.550.
- 14 Ricardo Lopes, Joao Magalhaes, and David Semedo. Glória: A generative and open large language model for portuguese. In *Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1*, pages 441–453, Santiago de Compostela, Galicia/Spain, March 2024. Association for Computational Linguistics. URL: <https://aclanthology.org/2024.propor-1.45/>.
- 15 Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting for retrieval-augmented large language models. *EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 5303–5315, May 2023. doi:10.18653/v1/2023.emnlp-main.322.
- 16 Yosi Mass, Boaz Carmeli, Haggai Roitman, and David Konopnicki. Unsupervised faq retrieval with question generation and bert. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 807–812, 2020. doi:10.18653/V1/2020.ACL-MAIN.74.
- 17 Bhaskar Mitra and Nick Craswell. An introduction to neural information retrieval. *Foundations and Trends® in Information Retrieval*, 13:1–126, December 2018. doi:10.1561/15000000061.
- 18 Vivi Nastase, Giuseppe Samo, Chunyang Jiang, and Paola Merlo. Multilingual vs. monolingual transformer models in encoding linguistic structure and lexical abstraction, 2025. URL: <https://aclanthology.org/2025.clicit-1.78/>.
- 19 Mindi Richia Putri, Ario Yudo Husodo, and Budi Irmawati. Simplification of embedding process in retrieval augmented generation for optimizing question answering chatbot model. *COMNETSAT 2024 - IEEE International Conference on Communication, Networks and Satellite*, pages 665–670, 2024. doi:10.1109/COMNETSAT63286.2024.10862926.
- 20 João Rodrigues, Luís Gomes, João Silva, António Branco, Rodrigo Santos, Henrique Lopes Cardoso, and Tomás Osório. Advancing neural encoding of portuguese with transformer albertina pt-*. In *Progress in Artificial Intelligence - 22nd EPIA Conference on Artificial Intelligence, EPIA 2023, Proceedings*, volume 14115 of *Lecture Notes in Artificial Intelligence*, pages 441–453, Faial Island, Azores, Portugal, September 2023. Springer. doi:10.1007/978-3-031-49008-8_35.
- 21 Mark D. Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. *International Conference on Information and Knowledge Management*, pages 623–632, 2007. doi:10.1145/1321440.1321528.
- 22 Hassan Soliman, Hitesh Kotte, Miloš Kravčik, Norbert Pengel, and Nghia Duong-Trung. Retrieval-augmented chatbots for scalable educational support in higher education. *CEUR Workshop Proceedings*, 2025. URL: <https://www.scopus.com/pages/publications/105011268364?origin=resultslist>.
- 23 Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *Advances in Neural Information Processing Systems*, April 2021. arXiv:2104.08663.
- 24 S. Vijayakumaran, M. Vimaladevi, R. Thangamani, N. Vibeesh, M. Chandru, and Sathishkumar Veerappampalayam Easwaramoorthy. Revolutionizing legal access: An ai-driven rag chatbot for real-time judicial insights. *Proceedings - 3rd International Conference on Artificial Intelligence and Machine Learning Applications: Healthcare and Internet of Things, AIMLA 2025*, 2025. doi:10.1109/AIMLA63829.2025.11040439.
- 25 Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, February 2024. arXiv:2212.03533.
- 26 Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, February 2024. doi:10.48550/arXiv.2402.05672.

- 27 Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. Miracl: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131, 2023. doi:10.1162/TACL_A_00595.
- 28 Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*, January 2021. arXiv:2101.00774.

Evaluating the Prosodic Diversity of TTS Models for L2 Prosody Assessment

Mariana Julião¹ 

HLT, INESC-ID, Lisbon, Portugal

Alberto Abad 

HLT, INESC-ID, Lisbon, Portugal

Instituto Superior Técnico, Lisbon, Portugal

Helena Moniz 

HLT, INESC-ID, Lisbon, Portugal

Faculdade de Letras da Universidade de Lisboa, Portugal

CLUL, Lisbon, Portugal

Abstract

Recent advances in text-to-speech (TTS) have led to synthetic speech that is often indistinguishable from natural speech at the level of individual utterances. However, it remains unclear whether such systems reproduce the prosodic variability observed in natural speech in a large native-speaker population. This question is particularly relevant for applications in computer-assisted language learning (CALL), as variability is a central property of prosody and a prerequisite for robust assessment.

In this work, we investigate whether TTS can approximate the distribution of prosodic patterns found in native speech, and whether it can serve as a reference for evaluating second language (L2) productions. To this end, we first compare different TTS models to native speakers, and then compare L2 speakers to synthetic speech by the model previously seen as the closest to native speakers.

Our results show that regardless of TTS achieving high perceptual quality, its prosodic variability substantially differs from that of native speakers. As a consequence, comparisons between L2 speech and TTS-based references reveal both the potential and the limitations of using synthetic speech for prosody assessment.

2012 ACM Subject Classification Applied computing → Computer-assisted instruction; Applied computing → Education

Keywords and phrases Speech synthesis, TTS, prosody, L1, L2

Digital Object Identifier 10.4230/OASICS.SLATE.2026.2

Funding Work supported by Portuguese national funds through Fundação para a Ciência e a Tecnologia, I.P. (FCT) under projects UID/50021/2025 (DOI: <https://doi.org/10.54499/UID/50021/2025>) and UID/PRR/50021/2025 (DOI: <https://doi.org/10.54499/UID/PRR/50021/2025>), by CLUL UID/214/2025 (DOI: <https://doi.org/10.54499/UID/00214/2025>) and by the Portuguese Recovery and Resilience Plan and NextGenerationEU European Union funds under project C644865762-00000008 (ACCELERAT.AI).

1 Introduction

Synthetic speech has reached a level of quality at which individual utterances are often indistinguishable from natural speech. Modern text-to-speech systems can generate highly natural signals across a range of voices and speaking styles. However, this success is typically

¹ Corresponding author



© Mariana Julião, Alberto Abad, and Helena Moniz;
licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 2; pp. 2:1–2:13

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

evaluated at the level of isolated utterances or individual speakers. It remains unclear whether these systems exhibit the same properties at a population level. In other words, when considered collectively, do multiple synthetic utterances with diverse timbres behave like a population of natural speakers, or do they reveal systematic biases and reduced variability?

The rapid development of TTS has already led to successful applications in second language learning, as reported in previous work [8, 14, 2, 20]. In these contexts, synthetic speech is primarily used as a production model: learners are exposed to target utterances and are encouraged to imitate them. While effective for pronunciation training, this paradigm does not address a complementary and equally important task, namely the assessment of learners' spontaneous productions.

In the field of CALL, prosody assessment for L2 speech remains a challenging problem. We identify three key limitations in current approaches.

First, many systems rely on imitation-based paradigms, where learners reproduce a reference utterance. While such approaches can be useful in specific cases, particularly for languages with markedly different prosodic systems, to which the learner needs to get acquainted (e.g., tonal languages), they do not capture a learner's ability to produce prosody in a natural communicative context. Mastery of prosody involves not only the correct realization of acoustic patterns, but also the ability to deploy them appropriately depending on linguistic and pragmatic context. Assessment frameworks should therefore move beyond imitation and support the evaluation of unconstrained productions.

Second, prosody exhibits inherent variability. A single sentence with a given communicative intent can be realized through multiple valid prosodic patterns. Effective assessment systems must therefore account for this one-to-many mapping between form and function, rather than enforcing a single canonical realization.

Finally, transparency is a critical requirement for language learning technologies. Learners benefit from feedback that is specific, interpretable, and actionable. Systems that rely solely on global scores or opaque quality metrics provide limited pedagogical value. Instead, prosody assessment should identify which aspects of a learner's production require improvement and how they relate to the target patterns.

We therefore argue that, if TTS systems can faithfully reproduce the variability of native human speech, they may provide a useful basis for overcoming the aforementioned limitations in L2 prosody assessment. Rather than relying on a single human reference or on fixed imitation-based tasks, synthetic speech could make it possible to generate multiple acceptable realizations of the same utterance, better reflecting the one-to-many nature of prosody. However, this potential depends on whether current TTS systems are sufficiently close to native-speaker populations, whether combining different systems can better approximate native variability, and whether synthetic speech can effectively replace human references when assessing learner productions.

The Research Questions (RQs) we address in this article are: (1) How closely does the prosodic diversity generated by different TTS models match that of a native speaker population? (2) Can the combination of multiple TTS models achieve prosodic variability that more closely matches that of a native speaker population? (3) Can synthetic speech be used in the place of human references for L2 prosody assessment?

2 Related Work

The use of TTS in Computer-Assisted Language Learning has been discussed since at least the late 2000s. Early work, such as that of Handley [8], examined its potential for learning French and concluded that, largely due to limited expressiveness, the available TTS systems were not yet suitable for such applications. A decade later, however, the outlook had become markedly more positive.

Bione et al. [1] evaluated the use of TTS for pronunciation instruction with native speakers of Brazilian Portuguese and found that, across most assessment measures, the synthetic voice performed comparably to a human voice, achieving high intelligibility. Similarly, Liakin et al. [14] investigated the use of TTS for learning French liaison and concluded that it constitutes a valuable complement to classroom instruction. Tejedor [18] employed TTS to provide corrective feedback on minimal pairs in a pronunciation training tool, showing that learners using the system significantly outperformed a control group. Overall, these studies primarily focus on segmental aspects of speech.

Other works using TTS included prosodic aspects. The study of Bonces et al. [2] focused on understanding the effect of TTS on the reading of L2. Results indicated a favourable effect, namely in aspects such as linking sounds, pronunciation accuracy, and reading timing. The work of Xiao et al. [20] intended to assess L2 prosody using a native speaker or a TTS reference, by comparing its F0 contours and breaks, to classify speakers as native or non-native. Results showed that the accuracy of the system trained using native speakers is 91.7%, while the accuracy of the system trained using TTS speech is 88.1%. This points towards TTS being sufficiently close to native speech to replace it for this purpose, given its practical advantages. A different application of TTS in prosody-related assessment is presented in the work of Korzekwa [13], who propose a system for pronunciation and lexical stress error detection. The authors start from the observation that annotated data containing lexical stress errors is often scarce, which limits effective model training. To address this issue, they use TTS to generate synthetic examples of such errors.

Hu et al. [11] have investigated the pragmatic capabilities of OpenAI's TTS across three pragmatic conditions: non-contrastive, contrastive, and corrective. The authors found that while TTS is able to produce a variety of F0 contours, these are diverse from the ones realized by humans in all three pragmatic conditions investigated.

3 Methodology

The present study comprises two main components. The first examines the prosodic variability of different TTS models. The second evaluates whether the model exhibiting the highest variability correlates more strongly with high-performing L2 speakers than with low-performing ones, in terms of prosody.

Our methodology involves generating synthetic speech and comparing it with native speech. This comparison is carried out using two approaches. The first computes the similarity between the prosodic features of each TTS utterance and those of all L1 utterances corresponding to the same prompt. The average similarity per speaker is then used as an indicator of prosodic distance from native English. The second approach defines a reference interval based on the prosodic features of L1 speakers and measures the proportion of TTS feature values that fall outside this interval.

For the second component, the methodology consists of comparing L2 utterances with TTS-based references. The validity of this approach is assessed by examining the extent to which the resulting scores align with the proficiency labels assigned to the L2 utterances.

3.1 Speech Synthesis

In this work, we compare three different TTS models: Coqui (XTTS v2 model)² [3], CosyVoice (Fun-CosyVoice 3.0 model)³ [4, 5] and VoxCPM (model 1.5)⁴ [23].

For each text, we generated 50 utterances, each leveraging a different reference audio from a unique VCTK (Section 4.1.2) utterance, thereby emulating the voice characteristics of distinct speakers and utterances. Reference audios were randomly sampled from the subset of recordings with a minimum duration of 6 seconds, as required by some of the models. The choice of VCTK is motivated by the fact that it includes a diversity of native speakers of English from different varieties. For CosyVoice, we also used the instruction modality, and randomly assigned a prosody-eliciting prompt from the list in Appendix A.

The synthesized utterances retained this diversity and naturalness. For the same text, the generated utterances exhibited a range of timbres, ages, styles, and dialectal variations. Even at the phonetic level, some synthesized outputs demonstrated pronounced rhoticity, while others did not. This variability is particularly valuable, as it enables the assessment of speakers without bias toward a single specific variety of English.

3.2 Features

We extracted prosodic features and applied perceptually motivated normalization techniques for energy and pitch. Word and phone boundaries were identified using the Montreal Forced Aligner (MFA) [15], which demonstrated robust results for L1, L2 and TTS speech. The type of speech representation considered is envisioned for the comparison of utterances of the same text (or of texts of the same prosodic structure). Therefore, it is expected that the n^{th} point of the feature vectors of two utterances compared corresponds to the same portion of the same word.

3.2.1 Pitch

To extract pitch, we used pyYAAAPT⁵ [22], which extracts F0 at 10 ms steps. We use pyYAAAPT for pitch extraction because it provides a robust implementation of the YAAAPT algorithm, which combines time-domain and frequency-domain information to estimate fundamental frequency. This hybrid approach makes it less sensitive to common pitch-tracking errors, such as octave jumps, voicing misclassification, and irregularities in the acoustic signal.

We aimed to follow the methodology of Elvira [6], which involved prosody comparison by extracting three points per vowel – at the beginning, middle, and end. However, this approach proved unfeasible in our case due to frequent vowel reductions, particularly in the speech of advanced speakers, which often precluded the collection of three distinct points. To address this, we opted to extract $N \times 3$ points per word as a proxy, where N represents the number of vowels. A potential advantage of this approach is that it includes voiced consonants, which carry relevant prosodic information, particularly through pitch-bearing sonorants such as nasals and liquids, as well as through duration and intensity cues.

Since our focus is on perceived pitch rather than absolute F0 values, we converted F0 measurements into semitones. This transformation provides a more accurate representation of perceived intonation and enables fair comparisons between utterances from speakers with

² <https://docs.coqui.ai/en/latest/models/xtts.html>

³ <https://huggingface.co/FunAudioLLM/Fun-CosyVoice3-0.5B-2512>

⁴ <https://github.com/OpenBMB/VoxCPM>

⁵ http://bjbschmitt.github.io/AMFM_decomp/pYAAAPT.html

different vocal pitch ranges. To compare the overall intonation of entire utterances, we applied utterance-wise pitch normalization by subtracting the lowest pitch value across the utterance.

3.2.2 Loudness

For loudness, we applied the same principles as for pitch. Specifically, we used the Loudness LLD from GeMAPS [7], which provides an “estimate of perceived signal intensity based on an auditory spectrum.” We chose loudness over other energy metrics to ensure relevance from a perceptual perspective. As with pitch, we extracted $N \times 3$ points per word, where N represents the number of vowels. Utterance-wise normalization was performed by dividing the loudness values by the maximum value within each utterance.

3.2.3 Duration

Using the alignment boundaries, we measured the duration of individual words as well as the intervals between them. We also examined phone- and vowel-level durations; however, word-level durations proved to be more informative.

3.3 Metrics

In the first part of this work, we assessed TTS models having L1 speakers as reference; in the second part, we assessed L2 speech having TTS speech as a reference. Therefore, in the next subsections, we refer to different types of utterances as test and reference.

3.3.1 Similarity Value

Each pair of utterances was compared using Pearson’s correlation coefficient, ρ , in an approach similar to that of Elvira-García et al. [6]. For each probe utterance utt_i , a value of Similarity S_i is computed as

$$S_i = \frac{1}{M} \sum_{j=1}^M \rho(utt_i, utt_j), \quad (1)$$

where M is the number of reference utterances utt_j with the same text as utt_i .

3.3.2 Reference Interval

We also computed a reference interval (RI) for each utterance and each feature, with the aim of characterizing the range of acceptable feature values for a given utterance. In a production setting, this metric could be used to indicate where and how a learner’s production diverges from the reference distribution. We adopted a non-parametric approach, defining the lower and upper limits of each RI as the 2.5th and 97.5th percentiles, respectively [12]. We define the Outlier Ratio R as the ratio of feature points falling within the RI. This ratio provides an additional measure of how closely the prosodic profile of a test utterance aligns with that of the reference utterances.

4 Experiments

4.1 Data

4.1.1 Speech Accent Archive

The Speech Accent Archive (SAA) [19] is a publicly accessible database that contains audio samples of speakers from various linguistic backgrounds reading the same scripted passage in English, the phonetically-rich paragraph *Please call Stella*.

Due to the large number of native English speakers reading this passage, the corpus enables the characterization of the range of acceptable realizations of a single text. We sampled 200 speakers to represent a native population.

4.1.2 VCTK

The *CSTR VCTK* corpus [21] is a corpus for voice cloning that comprises 108 English speakers, with average age of 22.7 years, of different varieties of English, with approximately 400 utterances per speaker. These correspond to a phonetically-rich paragraph, common to all speakers, and a newspaper text which is different for all speakers.

4.1.3 BNA

The Brave New Approach corpus (BNA) is a combination of the L2 corpora AUWL [9], ISLE [16] (train set), and C-AuDiT [10] (test set) specifically curated for the Paralinguistics Sub-Challenge of Interspeech 2015 [17]. AUWL and ISLE have been annotated with the same method: each speech file was annotated by five phoneticians with respect to its prosody (defined as sentence melody and rhythm) on a five-level scale, where 1 is *normal* for a native speaker and 5 is *very unusual* for a native speaker, which has then been averaged. The annotation of C-AuDiT, on the other hand, ranges from 0 to 2, where 0 is *good* and 2 is *bad*. The train set includes 3,980 utterances, while the test set includes 999 utterances. Only the train set is considered in the current work.

4.2 Model Comparison

We compared the TTS models by computing their Similarity and Outlier Ratio, having the native speech obtained from the SSA database as a reference. We also combined the utterances from different models and computed the Outlier Ratio, R. TTS models use the speakers of the VCTK corpus as target speakers to generate the samples.

4.3 TTS for L2 assessment

After selecting the TTS model or combination of models yielding the best results, we evaluate its potential for L2 prosody assessment. This is achieved by computing a similarity score for each L2 utterance with respect to the selected model, and subsequently comparing these scores to the corresponding prosodic labels using Spearman’s rank correlation coefficient. This metric has been widely used in prior work on the BNA corpus [17].

The experiments are conducted at both the utterance level and the aggregated label level. In the latter case, similarity scores are averaged across utterances sharing the same label.

■ **Table 1** Similarity values ($\uparrow$) between each model and the set of L1 speakers, for each feature. CosyVoice_P – CosyVoice with prosodic instructions.

Models	Coqui	CosyVoice	CosyVoice_P	VoxCPM
Pitch	0.540	0.541	0.527	0.476
Loudness	0.644	0.634	0.643	0.634
Duration	0.968	0.974	0.973	0.973
<i>Average</i>	0.718	0.717	0.714	0.694

■ **Table 2** Outlier ratio (R) ($\downarrow$) for each model or model combination, based on the set of L1 speakers, for each feature. Cq – Coqui; Cv – CosyVoice; Cvp – CosyVoice with prosodic instructions; VoxCPM – V.

Models	Cq	Cv	Cvp	V
Pitch	0.062	0.049	0.040	0.068
Loudness	0.046	0.128	0.125	0.094
Duration	0.029	0.008	0.015	0.025
<i>Average</i>	0.046	0.062	0.060	0.062

Models	Cq+Cv	Cq+Cvp	Cq+V	Cv+Cvp	Cv+V	Cvp+V
Pitch	0.049	0.040	0.068	0.040	0.068	0.068
Loudness	0.128	0.125	0.094	0.125	0.094	0.094
Duration	0.008	0.015	0.025	0.015	0.025	0.025
<i>Average</i>	0.062	0.060	0.062	0.060	0.062	0.062

Models	Cq+Cv+Cvp	Cq+Cv+V	Cq+Cvp+V	Cv+Cvp+V	All
Pitch	0.040	0.068	0.068	0.068	0.068
Loudness	0.125	0.094	0.094	0.094	0.094
Duration	0.015	0.025	0.025	0.025	0.025
<i>Average</i>	0.060	0.062	0.062	0.062	0.062

5 Results and Discussion

5.1 Model Comparison

The results of TTS model evaluation regarding similarity S values with L1 are in Table 1. The results of TTS model and model combination evaluation regarding Outlier Ratio R values with L1 are in Table 2. For clarity, in both cases, we also provide the average of the values of the three features considered.

The results indicate that, although certain models outperform others for specific features, overall performance differences remain relatively limited. For instance, Coqui achieves the best results for loudness, while CosyVoice performs best for duration.

We also observe that combining models, which increases the number of samples used to define the reference interval, does not generally lead to improved performance. Although combining outputs from different TTS systems led to small changes in the Outlier Ratios, these differences did not indicate a consistent or substantial improvement over the best individual systems. This suggests that aggregating several generative models may increase the number of synthetic samples without substantially expanding the prosodic space in the dimensions that matter for native-like variability. In other words, model diversity at the system level does not automatically translate into prosodic diversity at the acoustic-feature level.

■ **Table 3** Spearman’s correlation coefficient between similarity scores S and prosody labels, for utterance level and aggregated-label level. p -values are represented by asterisks according to significance: * : $p > 5 \times 10^{-2}$; ** : $1 \times 10^{-5} < p \leq 5 \times 10^{-2}$; *** : $p \leq 1 \times 10^{-5}$.

	Utterance Level	Aggregated Label Level
Pitch	0.062**	0.745**
Loudness	0.214***	0.973***
Duration	0.205***	0.982***

Furthermore, CosyVoice with prosodic instruction stands out only in achieving the highest correlation for pitch, with no comparable gains for other features. Notably, the best overall average results are obtained with Coqui, the oldest model considered in this study. This finding cautions against assuming that more recent or larger TTS models are necessarily better suited for modelling native prosodic variability. One possible explanation is that newer systems may produce more stable or regularized outputs, reflecting optimization for naturalness, intelligibility, or speaker consistency rather than for variability across acceptable prosodic realizations. Coqui’s comparatively stronger performance may therefore result from its preserving a wider range of acoustic variation, especially in energy-related dimensions. Nevertheless, this explanation should be interpreted cautiously, since the models differ in multiple respects, including architecture, training data, synthesis pipeline, and default generation settings.

These findings suggest that, despite substantial advances in TTS quality, recent models do not appear to improve in terms of prosodic variability. Instead, progress seems to be primarily driven by improvements in the quality of individual utterances rather than in the diversity of their realizations.

5.2 TTS for L2 assessment

To address whether synthetic speech can be used as a reference for L2 prosody assessment, we selected Coqui, the best-performing model in the previous experiments. We evaluated the relationship between similarity scores (S) – computed via the average Pearson’s correlation between TTS and L2 utterances – and the human-assigned prosodic labels. This was computed using Spearman’s correlation coefficient, and assessed both at the utterance level and at the aggregated label level. Results are in Table 3.

On an utterance-by-utterance basis, the correlation between TTS similarity and proficiency was relatively low. For loudness and duration, correlation values hovered around 0.2, while pitch showed almost no significant relationship (near 0.0). This indicates that a high similarity to a TTS reference does not always guarantee a high human rating for a specific utterance.

When similarity scores were aggregated by label (averaging scores for all utterances of a given label), the correlations became strikingly strong. For loudness and duration, the absolute Spearman’s rank correlation reached 0.973 and 0.982, respectively. For pitch, it reached 0.745. This suggests that, while TTS cannot reliably assess the quality of individual utterances, it accurately captures the general prosodic trends associated with increasing L2 proficiency.

■ **Table 4** Average Similarity Value (S) between L2 utterances and TTS references, and number of outlier points ($\#$ Outliers) to the reference interval, for Utt1.0, Utt2.4 and Utt3.2.

		Pitch	Loudness	Duration
S	Utt1.0	0.478	0.800	0.969
	Utt2.4	0.617	0.593	0.968
	Utt3.2	0.346	0.817	0.950
# Outliers	Utt1.0	8	5	3
	Utt2.4	0	12	6
	Utt3.2	4	2	8

5.3 Reference Intervals for Local Analysis of L2 Speech

To better understand the results previously presented, we focus on utterances corresponding to the prompt *I'd like a one-way ticket to London, please*: one with highest rating (Utt1.0), one with average-good rating (Utt2.4) and one with average-low rating (Utt3.2). We plotted the feature sequences alongside the TTS references in Figure 1. Table 4 presents the values of Average Similarity and $\#$ Outliers for these utterances.

For pitch, Utt1.0 has 8 outliers: twice as much as Utt3.2. Utt2.4, however, does not have any outliers. The different sequences and the centroid appear to have mostly the same monotony – Utt1.0's outliers being mostly due to major shifts upward and downward, which can be seen as just an emphasized intonation by a proficient speaker. Utt3.2, nevertheless, presents jumps that are in the opposite direction as the rest, which is mostly visible right at the beginning.

For loudness, however, Utt2.4 is the one with the most outliers, which often correspond to an exaggeration of the tendency peaks of the distribution. Utt3.2 yields the best performance having the lowest number of outliers.

For duration, the number of outliers correlates to the labels. The fact that the intervals between words are considered allows for differentiation between good and worse sequences – normally, due to hesitations, worse speakers can spend more time between words. This is the case of Utt3.2 between the words *London* and *please*. However, this example seems to also confirm the idea that less proficient speakers have a lower speech rate and spend more time uttering words, as pointed out by the outliers at the top of the plot.

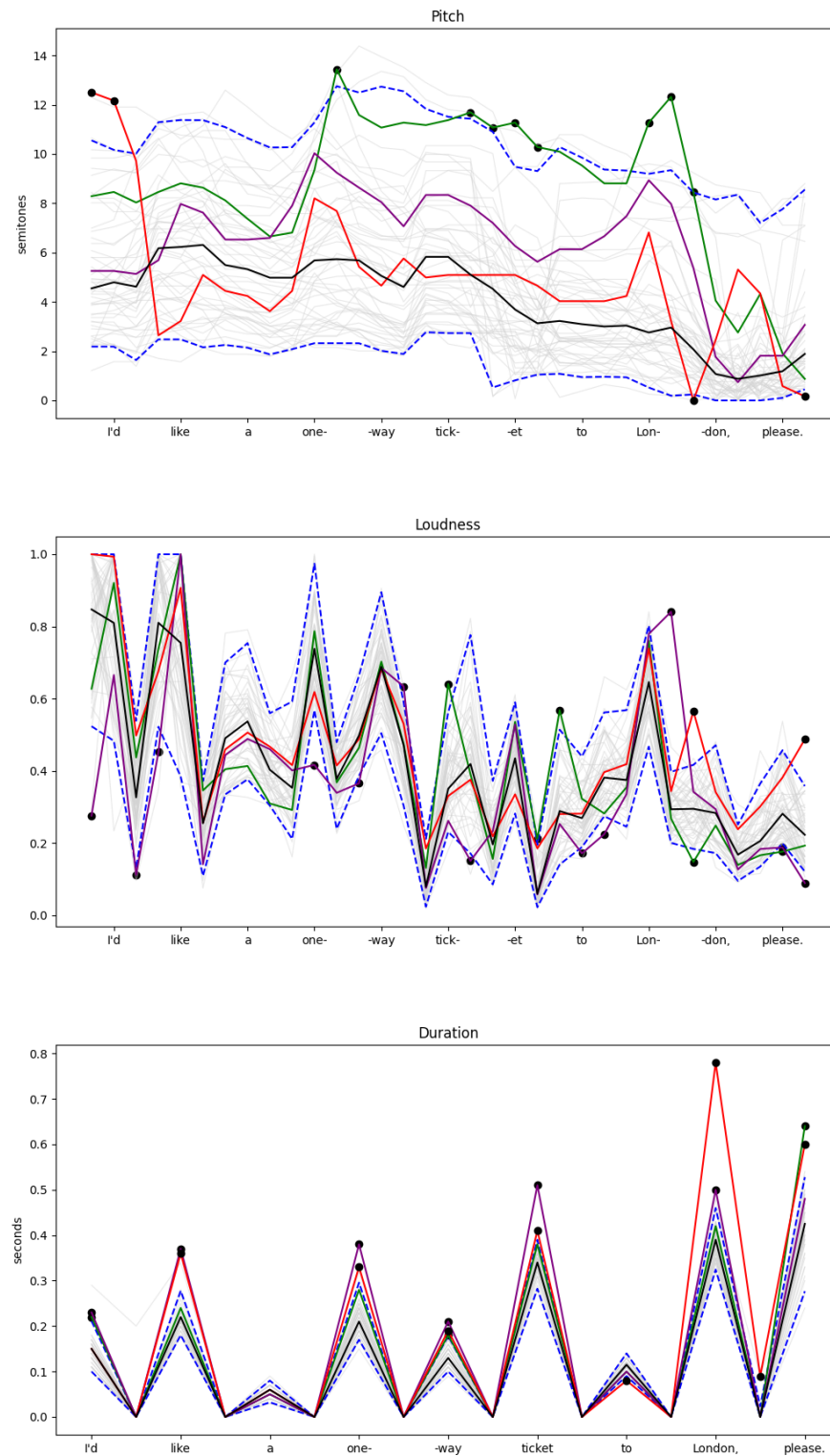
6 Conclusion

We first examined the similarity between the prosodic feature distributions of different TTS models and those of native speakers. We then evaluated the potential of the best-performing model to serve as a reference for L2 prosody assessment.

Our results show that trends observed at the aggregate level do not consistently hold at the utterance level. Individual utterances with higher proficiency scores do not systematically exhibit stronger alignment with the TTS distribution, nor do they consistently present fewer deviations from the reference interval. A more detailed analysis further reveals that local behavior is highly variable, limiting the reliability of TTS-based comparisons at the utterance level.

These findings are consistent with our qualitative observations. While synthetic speech exhibits some variability, it also displays noticeable regularities across utterances generated from the same prompt. In contrast, low-scoring L2 productions tend to diverge substan-

2:10 Evaluating the Prosodic Diversity of TTS Models for L2 Prosody Assessment



■ **Figure 1** Pitch, Loudness and Duration sequence representation for prompt *I'd like a one-way ticket to London, please.* – TTS samples; – **centroid**; - - 2.5% and 97.5% reference intervals; – Utt1.0; – Utt2.4; – Utt3.2; ● point falling outside reference interval.

tially from synthetic speech, whereas high-scoring productions, although generally closer to TTS, often achieve correctness through patterns that differ from the synthesized trend. In some cases, high-scoring utterances even exaggerate these patterns, resulting in outliers, as illustrated by the pitch distribution in Figure 1.

With respect to our research questions, we find that (RQ1) different TTS models do not differ substantially in their overall proximity to native speech, although individual models may perform better for specific prosodic features. Regarding RQ2, combining outputs from multiple models does not increase prosodic diversity in a way that more closely approximates native variability. Finally, for RQ3, our results suggest that synthetic speech cannot yet reliably replace human references for L2 prosody assessment, as its limitations at the utterance level undermine its effectiveness for this purpose.

We argue that these limitations stem from the design objectives of current TTS systems. Such systems are primarily optimized for perceptual naturalness at the level of individual utterances, rather than for capturing the full range of variability present in natural speech. Consequently, the variability observed in synthetic speech appears to be an incidental by-product rather than a controlled or representative feature.

References

- 1 Tiago Bione, Jennica Grimshaw, and Walcir Cardoso. An evaluation of TTS as a pedagogical tool for pronunciation instruction: the ‘foreign’ language context. *CALL in a climate of change: adapting to turbulent global conditions—short papers from EUROCALL*, pages 56–61, 2017.
- 2 Jeisson Alonso Rodríguez Bonces. Implementing text-to-speech technology as a means of enhancing L2 reading fluency. *EDU REVIEW. International Education and Learning Review/Revista Internacional de Educación y Aprendizaje*, 1(2):61–73, 2019.
- 3 Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Gökner, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, et al. XTTS: a Massively Multilingual Zero-Shot Text-to-Speech Model. *arXiv preprint arXiv:2406.04904*, 2024. [arXiv:2406.04904](#).
- 4 Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, et al. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*, 2024. [arXiv:2407.05407](#).
- 5 Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Xian Shi, Keyu An, et al. CosyVoice 3: Towards In-the-wild Speech Generation via Scaling-up and Post-training. *arXiv preprint arXiv:2505.17589*, 2025. [arXiv:2505.17589](#).
- 6 Wendy Elvira-García, Simone Balocco, Paolo Roseano, and Ana Ma Fernandez-Planas. ProDis: A dialectometric tool for acoustic prosodic data. *Speech Communication*, 97:9–18, 2018. [doi:10.1016/J.SPECOM.2017.12.013](#).
- 7 Florian Eyben, Klaus R Scherer, Björn W Schuller, Johan Sundberg, Elisabeth André, Carlos Busso, Laurence Y Devillers, Julien Epps, Petri Laukka, Shrikanth S Narayanan, et al. The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing. *IEEE transactions on affective computing*, 7(2):190–202, 2015. [doi:10.1109/TAFFC.2015.2457417](#).
- 8 Zöe Handley. Is Text-to-Speech synthesis ready for use in Computer-Assisted Language Learning? *Speech Communication*, 51(10):906–919, 2009. [doi:10.1016/J.SPECOM.2008.12.004](#).
- 9 Florian Hönig, Anton Batliner, and Elmar Nöth. Automatic assessment of nonnative prosody—annotation, modelling and evaluation. In *International Symposium on Automatic Detection of Errors in Pronunciation Training (IS ADEPT)*, pages 21–30, 2012.

- 10 Florian Hönig, Anton Batliner, Karl Weilhammer, and Elmar Nöth. Islands of Failure: Employing word accent information for pronunciation quality assessment of English L2 learners. In *International Workshop on Speech and Language Technology in Education*, 2009.
- 11 Na Hu, Jiseung Kim, Riccardo Orrico, Stella Gryllia, and Amalia Arvaniti. Can OpenAI's TTS model convey information status using intonation like humans? In *Speech Prosody 2024*, pages 32–36, 2024. doi:10.21437/SpeechProsody.2024-7.
- 12 Kiyoshi Ichihara, James C Boyd, IFCC Committee on Reference Intervals, and Decision Limits (C-RIDL). An appraisal of statistical procedures used in derivation of reference intervals. *Clinical chemistry and laboratory medicine*, 48(11):1537–1551, 2010.
- 13 Daniel Korzeczkwa, Jaime Lorenzo-Trueba, Thomas Drugman, and Bozena Kostek. Computer-assisted pronunciation training — Speech synthesis is almost all you need. *Speech Communication*, 142:22–33, 2022. doi:10.1016/J.SPECOM.2022.06.003.
- 14 Denis Liakin, Walcir Cardoso, and Natallia Liakina. The pedagogical use of mobile speech synthesis (TTS): Focus on French liaison. *Computer Assisted Language Learning*, 30(3-4):325–342, 2017.
- 15 Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. In *Proc. Interspeech 2017*, pages 498–502, 2017. doi:10.21437/Interspeech.2017-1386.
- 16 Wolfgang Menzel, Eric Atwell, Patrizia Bonaventura, Daniel Herron, Peter Howarth, Rachel Morton, and Clive Souter. The ISLE corpus of non-native spoken English. In *Proceedings of LREC 2000: Language Resources and Evaluation Conference, vol. 2*, pages 957–964. European Language Resources Association, 2000.
- 17 Björn Schuller, Stefan Steidl, Anton Batliner, Simone Hantke, Florian Hönig, J. R. Orozco-Arroyave, Elmar Nöth, Yue Zhang, and Felix Weninger. The INTERSPEECH 2015 computational paralinguistics challenge: nativeness, Parkinson's & eating condition. In *Interspeech 2015*, pages 478–482, 2015. doi:10.21437/Interspeech.2015-179.
- 18 Cristian Tejedor-Garcia, David Escudero-Mancebo, César González-Ferreras, Enrique Cámara-Arenas, and Valentín Cardenoso-Payo. Evaluating the efficiency of synthetic voice for providing corrective feedback in a pronunciation training tool based on minimal pairs. *Proc. SLaTE, Stockholm, Sweden*, pages 26–30, 2017.
- 19 Steven Weinberger. Speech Accent Archive. Retrieved from <http://accent.gmu.edu>, 2015.
- 20 Yujia Xiao and Frank K Soong. Proficiency Assessment of ESL Learner's Sentence Prosody with TTS Synthesized Voice as Reference. In *INTERSPEECH*, pages 1755–1759, 2017. doi:10.21437/INTERSPEECH.2017-64.
- 21 Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92), 2019. [sound]. doi:10.7488/ds/2645.
- 22 Stephen A Zahorian and Hongbing Hu. A spectral/temporal method for robust fundamental frequency tracking. *The Journal of the Acoustical Society of America*, 123(6):4559–4571, 2008.
- 23 Yixuan Zhou, Guoyang Zeng, Xin Liu, Xiang Li, Renjie Yu, Ziyang Wang, Runchuan Ye, Weiyue Sun, Jiancheng Gui, Kehan Li, Zhiyong Wu, and Zhiyuan Liu. VoxCPM: Tokenizer-Free TTS for Context-Aware Speech Generation and True-to-Life Voice Cloning. *arXiv preprint arXiv:2509.24650*, 2025. doi:10.48550/arXiv.2509.24650.

A List of Prosodic Prompts

1. Speak with quiet relief, as if a problem has just been resolved
2. Speak with mild frustration, while trying to remain polite
3. Speak with genuine enthusiasm and positive energy
4. Speak with a calm and reassuring tone
5. Speak with subtle anxiety beneath otherwise neutral wording
6. Speak with restrained excitement, trying not to overreact

7. Speak with quiet disappointment, keeping emotions controlled
8. Speak with gentle amusement, as if something is slightly funny
9. Speak with a serious and focused tone
10. Speak with a relaxed and informal attitude
11. Speak as if carefully choosing your words to avoid misunderstanding
12. Speak as if thinking aloud, including slight natural hesitations
13. Speak as if recalling something from memory with slight uncertainty
14. Speak with clear confidence and decisiveness
15. Speak as if double-checking your own reasoning while talking
16. Speak with slight hesitation, as if unsure about the statement
17. Speak as if explaining something you have just understood
18. Speak with a reflective and thoughtful tone
19. Speak with a sense of mild confusion
20. Speak with deliberate and precise articulation
21. Speak as if explaining something to a child
22. Speak as if giving instructions in a clear and direct way
23. Speak as if making a polite request
24. Speak as if defending a point in a discussion
25. Speak as if sharing a secret
26. Speak as if telling a short anecdote
27. Speak as if emphasizing an important point
28. Speak as if downplaying the importance of the message
29. Speak as if offering a suggestion rather than a command
30. Speak as if making a careful clarification
31. Speak as if talking to a close friend
32. Speak as if talking to someone you do not know well
33. Speak as if addressing a group of people
34. Speak as if speaking to someone you respect
35. Speak as if trying to be especially polite
36. Speak as if trying not to offend the listener
37. Speak with a slightly formal and professional tone
38. Speak with a warm and friendly tone
39. Speak with a neutral and detached tone
40. Speak with a slightly playful tone
41. Speak in a quiet environment, using a soft voice
42. Speak in a noisy environment, projecting your voice clearly
43. Speak as if you are slightly out of breath
44. Speak as if you are tired but still engaged
45. Speak with steady rhythm and even pacing
46. Speak with slightly faster-than-normal tempo
47. Speak with slightly slower-than-normal tempo
48. Speak with clear emphasis on key words
49. Speak with reduced emphasis, keeping the tone relatively flat
50. Speak with natural variation in pitch and rhythm

Cross-Language Text Readability Assessment: Leveraging Multilingual Models for Improved Performance in CEFR-Level Classification

Eugénio Ribeiro ✉ 

INESC-ID Lisboa, Portugal

Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, Lisboa, Portugal

Jorge Baptista ✉ 

INESC-ID Lisboa, Portugal

Faculdade de Ciências Humanas e Sociais, Universidade do Algarve, Faro, Portugal

Abstract

The automatic assessment of text readability and the classification of texts by levels is essential for language education and language industries that rely on effective communication. This study explores cross-language automatic readability level classification using the levels defined by the Common European Framework of Reference for Languages (CEFR). We investigate the potential of using data in one language to improve classification performance in different languages and, thus, optimize the utilization of the limited labeled resources available for each language. We rely on a pre-trained multilingual Transformer-based language model, by fine-tuning it on annotated data in one language or in a combination of languages, and then assessing its ability to generalize even to unseen languages. In an additional scenario, we further fine-tune the models on data in the target language, to assess whether the models trained on data in different languages can capture generic information regarding text readability and then be further specialized to capture the specific characteristics of the target language. Our experiments covering the English, Dutch, and German languages revealed that direct generalization to unseen languages is challenging. However, when paired with data in the target language, multilingual data can be leveraged to capture cross-language aspects of text readability, leading to more robust and better-performing models.

2012 ACM Subject Classification Computing methodologies → Natural language processing; Applied computing → Education

Keywords and phrases Readability, Text Complexity, CEFR, Multilinguality

Digital Object Identifier 10.4230/OASICS.SLATE.2026.3

Funding This work was supported by Portuguese national funds through Fundação para a Ciência e a Tecnologia, I.P. (FCT) under projects UID/50021/2025 (DOI: <https://doi.org/10.54499/UID/50021/2025>), UID/PRR/50021/2025 (DOI: <https://doi.org/10.54499/UID/PRR/50021/2025>), and UID/04466/2025 (DOI: <https://doi.org/10.54499/UID/04466/2025>); and by the European Commission (Project: iRead4Skills, Grant number: 1010094837, Topic: HORIZON-CL2-2022-TRANSFORMATIONS-01-07, DOI: <https://doi.org/10.3030/101094837>).

1 Introduction

The Common European Framework of Reference for Languages (CEFR) [9] provides a widely recognized framework for classifying language proficiency levels, ranging from A1 (beginner) to C2 (proficient). Additionally, the CEFR has been used to assess the proficiency of individuals as learners of different languages. However, it can also be used from a readability perspective, shifting the focus to understanding the proficiency required to read specific pieces of text. Identifying the readability level of a text is relevant across diverse domains, encompassing not only language education but also various language-related industries and many other human activities. In education, assessing the readability level allows educators



© Eugénio Ribeiro and Jorge Baptista;

licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 3; pp. 3:1–3:13

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and curriculum designers to match texts to the learners’ abilities, fostering effective language development and personalized learning experiences. Moreover, outside the education domain, readability level classification finds applications in different sectors. For instance, in the banking industry, presenting financial information and policies at an appropriate readability level ensures that clients can understand terms and conditions, enabling well-informed decision-making. Similarly, in healthcare, accessible and understandable medical instructions, consent forms, and patient information materials are crucial for individuals with varying levels of language proficiency. Furthermore, legal information, government communications, user manuals, and many others, benefit from accurately assessing the readability level of written materials, facilitating effective communication, content transparency, and general comprehension. Therefore, by exploring the readability perspective of the CEFR, we can make a significant contribution to enhancing the understanding of text comprehension factors and their far-reaching implications for both education and language-related industries, seeking to convey information to learners or clients in a manner that is clear, concise, and easily understood.

Determining the readability level of texts presents its own set of challenges, particularly when working with languages that have limited annotated resources. Annotating large amounts of text data with CEFR levels is a labor-intensive and time-consuming task, often requiring expert domain knowledge. Consequently, the scarcity of labeled data in multiple languages hinders the development of robust and accurate models for automatic readability level classification. To address this challenge, we look into the problem from a cross-language perspective, which offers a promising avenue for leveraging the limited annotated resources available for each language.

Starting with a pre-trained multilingual Transformer-based language model [48], we explore its fine-tuning for CEFR-level classification on data in three different languages – English, Dutch, and German – and assess the ability of the obtained fine-tuned models to generalize across languages. The aim of this study is not to achieve the highest possible performance on text readability level classification, but rather to assess whether data in one language can help improve the classification performance in different languages. This point is highly relevant. If data from various languages can be utilized to enhance performance in a specific target language, it suggests the existence of shared linguistic patterns and structures that exhibit a similar perceived difficulty level across languages. Moreover, this indicates that the limited availability of labeled resources can be effectively combined and leveraged in future model development for automatic readability level classification.

In the following sections, we will proceed as follows: Section 2 will provide an overview of related work on automatic text readability level classification. Then, in Section 3, we will describe our experimental setup, including the datasets used and the methodology employed for training and evaluating various models. Next, Section 4 will present and discuss the findings from our experiments. Finally, in Section 5, we will summarize the contributions of this study, discuss its limitations, and suggest directions for future research in this area.

2 Related Work

Readability assessment is a problem that has been widely explored over the years. Traditionally, the problem is addressed by creating readability formulas or indexes based on statistical information and/or domain knowledge [13, 10]. Among these, the most widely used are the Flesch Reading Ease index and the Flesch-Kincaid Grade Level [23].

However, considering the developments in Machine Learning (ML), and especially in Natural Language Processing (NLP), part of the research on automatic readability assessment shifted towards following the trends in the NLP area [33]. Early approaches (and many recent ones for low-resource languages) relied on handcrafted features, such as word frequency, sentence length, and syntactic complexity, combined with traditional machine learning algorithms, such as decision trees and support vector machines (e.g. [4, 16, 22, 11, 34, 15, 3]). Then, Deep Learning (DL) approaches relying on pre-trained word embeddings, such as those generated by Word2Vec [30], emerged (e.g. [8, 32, 14]). More recently, research in the area shifted towards the fine-tuning of pre-trained Transformer-based language models, such as BERT [12], GPT [35], and RoBERTa [27] (e.g. [44, 53, 28, 31, 40]). Still, some studies revealed that model fine-tuning and handcrafted features are actually complementary for the task and that the best performance can be achieved by combining both in hybrid models (e.g. [25, 26, 50, 1]). Prompt-based approaches relying on the intrinsic knowledge of Large Language Models (LLMs), such as those of the GPT [7], LLaMA [47], Gemma [18], and EuroLLM [29] families, have also been explored in both zero-shot and few-shot settings. However, they tend to underperform in comparison to fine-tuned approaches (e.g. [39, 20, 2]).

The previous studies on automatic text readability level assessment cover several languages, such as English (e.g. [52, 8, 32, 14, 28]), French (e.g. [16, 17, 53, 49, 19, 2]), Chinese (e.g. [46]), German (e.g. [31]), Brazilian (e.g. [4, 24]) and European Portuguese (e.g. [5, 11, 44, 24, 40, 42, 3]), Italian (e.g. [15, 45]), Russian (e.g. [22, 38]), Swedish (e.g. [21, 34]), Spanish (e.g. [1, 36]), Galician (e.g. [37]), and Slovenian (e.g. [28]). In this context, it is important to mention the UNIVERSALCEFR initiative [20], which aggregated CEFR-annotated data in 13 languages and explored several automatic assessment approaches. However, most of the languages are still under-resourced and, although several languages are covered in the studies, to the best of our knowledge, cross-language models have not been explored for the task. Thus, our study fills a gap that may define a path for future research in this area.

3 Experimental Setup

In this section, we describe our experimental setup. As we intend to explore cross-language text readability assessment, we need datasets in multiple languages. These are described in Section 3.1. Then, in Sections 3.2 and 3.3, we describe our training and evaluation procedures. Finally, in Section 3.4, we provide implementation details that enable the future reproduction of our experiments.

3.1 Datasets

In our experiments, we use the datasets collected in the context of the CEFR Labelling and Assessment Services (CEFRSERV) project [6]. These cover three languages – English (EN), Dutch (NL), and German (DE). We opted for using these datasets as they were collected in the context of the same project and cover languages of the same family – west-germanic –, which provides an interesting starting point for this cross-lingual analysis. Still, as the datasets were collected by different teams, some differences can be found in the annotation and the format in which the annotated texts are provided. In order to improve reproducibility, we describe these differences below and discuss how we address them in our experiments.

Firstly, the German dataset is annotated with the six CEFR levels, between A1 and C2, while the English and Dutch datasets also include intermediate levels, such as A1+ or B1–B2. For consistency and simplicity, we do not consider intermediate levels in this study and the texts annotated with those levels are considered as belonging to the lower level. The only exception is the A1- level, which only occurs in the English dataset and is considered as A1.

■ **Table 1** Level distribution of texts across datasets.

Language	A1	A2	B1	B2	C1	C2	Total
English	40	154	216	174	97	69	750
Dutch	31	155	520	410	73	7	1,196
German	31	90	90	115	88	95	509
Total	102	399	826	699	258	171	2,455

Also, the annotation of each text in the English and German datasets refers to an aggregate of multiple annotators, while the Dutch dataset includes all the annotations for each text. For consistency, we aggregate the latter as an average.

Finally, while the German and Dutch datasets are provided in a single file featuring all the text-annotation pairs, the English dataset is provided as multiple files. Besides, the dataset description refers to approximately 1,200 texts, but there are only 750 files. This means that some files contain more than one text with the same CEFR level. However, there is no straightforward way to automatically identify these files nor to separate the texts. Thus, we considered each file to be a single text.

Table 1 shows the distribution of the texts across languages and CEFR levels. We can see that there is a bias towards the B levels. This was also observed on many of the datasets used in previous research on automatic text readability level classification (e.g. [11, 38, 34, 15]). On the German dataset, this bias is not as pronounced as in the other two languages. However, this is also the smallest dataset, with only 509 annotated texts. Overall, there are 2,455 annotated texts, most of which (48.72%) are in Dutch.

3.2 Training Methodology

The aim of this study is to assess whether data in one language can help improve automatic text readability classification performance in different languages and, thus, whether scarce labeled resources can be used more efficiently.

To do so, we use as a base a pre-trained multilingual Transformer-based language model and then train multiple models by fine-tuning it for CEFR-level classification on data in each of the three languages and on each combination of those languages. The performance of these models can then be assessed across languages.

In our experiments, we explored two training approaches: single-step and two-step. In single-step training, each model is obtained by fine-tuning the base model on the training data of the corresponding language or combination of languages. This allows the assessment of the direct generalization ability of the models across languages. In two-step training, each of the models generated using single-step training is further fine-tuned on the training data for each of the target languages. In this scenario, the first training step can be seen as an adaptation of the base pre-trained model to the text readability classification task in general, while the second step is used to further refine the models so as to capture the particular characteristics of the target language in the context of the task.

Furthermore, in line with other studies on automatic CEFR-level classification (e.g., [11]), we train models for both six levels (A1–C2) and three levels (A–C). This approach enables us to assess whether a classification with lower granularity can leverage the information available from data in other languages more effectively.

■ **Table 2** Accuracy (Acc.) and adjacent accuracy (Adj. Acc.) results (%) using single-step training for six CEFR levels.

	English		Dutch		German	
	Acc.	Adj. Acc.	Acc.	Adj. Acc.	Acc.	Adj. Acc.
EN	62.89±3.01	97.11±1.39	43.06±0.64	94.31±0.24	34.64±4.84	80.06±0.57
NL	23.78±2.70	74.22±4.53	64.58±1.10	99.17±0.42	25.16±5.40	59.80±7.40
DE	26.22±5.43	65.11±10.21	18.19±4.45	54.17±5.12	67.32±1.50	87.91±4.42
EN+NL	62.67±3.06	96.67±1.34	64.86±0.86	99.86±0.24	27.78±4.53	68.30±1.50
EN+DE	62.89±2.14	97.33±0.00	26.53±1.58	79.44±0.24	63.72±1.70	89.22±2.60
NL+DE	24.00±0.67	70.22±1.02	63.75±2.21	99.44±0.24	65.69±3.93	86.93±1.14
All	59.33±3.06	97.11±0.77	65.00±3.25	99.44±0.24	65.36±2.47	89.55±1.13

3.3 Evaluation Methodology

Regarding evaluation metrics, we adopt *accuracy* and *adjacent accuracy*, which are consistently used in previous studies on automatic CEFR readability level classification (e.g., [16, 11, 53, 41, 1]). Accuracy evaluates the precise identification of a text’s readability level, while adjacent accuracy also considers neighboring levels, offering further insight into the identification of texts slightly easier or harder than the assigned level. As the coarse-grained setting considers only three levels, we rely solely on *accuracy* for assessing the performance in that context. Both metrics are reported as percentages.

Due to the absence of a standard partition in the datasets, we conduct a random 80/20 train/test split. To enhance robustness and mitigate the impact of randomness, we perform three independent experimental runs, each with different random seeds for the splitting process. The evaluation metrics are reported as both the average and standard deviation across these runs.

3.4 Implementation Details

As our objective is a comparative analysis rather than achieving the utmost performance on the task, we use the multilingual cased version of DistilBERT [43] as a base model. This choice enables faster experiments and reduces resource consumption. Each model undergoes 50 epochs of training and the weights from the best epoch are loaded in the end. For model training and evaluation, we rely on the functionality offered by HuggingFace’s Transformers library [51].

4 Results

In this investigation, we adopt a two-step training approach that builds on the models obtained through single-step training. Thus, to begin, in Section 4.1, we present and discuss the results obtained using the single-step training approach. Next, in Section 4.2, we focus on the results achieved with the two-step approach. Finally, in Section 4.3, we produce an overall discussion, draw further conclusions, and address certain limitations encountered during the study.

■ **Table 3** Accuracy (%) results using single-step training for three CEFR levels.

	English	Dutch	German
EN	79.11±1.92	72.50±3.63	52.94±1.96
NL	45.11±13.04	84.72±1.58	42.16±8.03
DE	48.44±1.39	49.03±12.31	81.04±5.74
EN+NL	81.11±2.04	85.28±0.48	46.08±1.96
EN+DE	79.56±0.77	61.53±4.09	79.08±2.04
NL+DE	55.78±7.73	84.58±1.91	76.47±5.88
All	78.22±2.04	85.00±2.21	78.43±3.53

4.1 Single-Step Training

The accuracy and adjacent accuracy results obtained through single-step training, considering the six CEFR levels, are presented in Table 2. Notably, the models demonstrate limited generalization capabilities on unseen languages. The highest achieved accuracy is 43.06% when applying the model trained on English texts to Dutch. This performance stands 20.69 percentage points below the lowest-performing model trained on Dutch data and 12.27 percentage points lower than the least performing model trained on the target language in general. These discrepancies may arise due to various factors, including distinct annotation guidelines for different languages, variations in nature of the text, or potential misalignment of cross-language representations in the base model. Nonetheless, these findings suggest a significant language dependence concerning the perceived difficulty level of texts.

Upon examining the model performance for those trained in the target language, we observe that the models trained in multiple languages demonstrate competitiveness with those trained exclusively on the target language. Remarkably, the top-performing model for Dutch results from training on the three languages, albeit with a higher degree of variability. This finding indicates the presence of cross-language aspects influencing text readability, and data from different languages can contribute to training more robust models. This observation is further supported by the adjacent accuracy results, which show that models trained on data from multiple languages consistently achieve the highest performance across all languages. An intriguing outcome pertains to the adjacent accuracy for German, which, despite having the highest accuracy, exhibits the lowest adjacent accuracy. This suggests certain challenges in capturing aspects of German texts' difficulty or potential issues with the annotation process, as the errors tend to be more severe.

Finally, as displayed in Table 3, when considering only three proficiency levels, the overall accuracy increases, exceeding 80% for every language, as anticipated. Notably, the model trained on data from both English and Dutch achieves the highest performance for both languages, while the model trained solely on German data remains the top performer for German. These results align with the high adjacent accuracy observed when considering the six CEFR levels for the models trained on English and Dutch data. Regarding adjacent accuracy, as mentioned in Section 3.3, the results for the three-level scenario are omitted, as the metric lacks informativeness in this context. In fact, every model trained on data from the target language achieves 100% adjacent accuracy, while the remaining models achieve at least 95%.

■ **Table 4** Accuracy (Acc.) and adjacent accuracy (Adj. Acc.) results (%) using two-step training for six CEFR levels.

	English		Dutch		German	
	Acc.	Adj. Acc.	Acc.	Adj. Acc.	Acc.	Adj. Acc.
EN	62.89±3.01	97.11±1.39	64.58±0.83	99.17±0.00	65.03±2.26	86.93±2.26
NL	64.67±2.67	96.89±0.38	64.58±1.10	99.17±0.42	65.69±0.98	86.60±2.26
DE	61.33±2.40	95.78±2.34	65.97±1.34	99.31±0.24	67.32±1.50	87.91±4.42
EN+NL	63.56±3.67	97.11±1.02	65.83±1.10	99.72±0.24	62.75±4.27	90.52±3.00
EN+DE	66.44±1.54	97.78±0.77	66.39±0.96	99.58±0.42	68.95±2.04	89.87±3.00
NL+DE	64.67±0.67	96.22±1.02	65.28±0.48	99.72±0.48	66.99±2.04	87.58±1.13
All	62.00±3.06	97.56±0.77	65.28±2.51	99.31±0.24	66.99±2.83	88.89±1.50

4.2 Two-Step Training

Table 4 displays the accuracy and adjacent accuracy results achieved through two-step training, considering the six CEFR levels. All models demonstrate competitive performance with the model trained solely on the target language in terms of accuracy. However, when analyzing the results for Dutch and German, we observe distinct behavior. For Dutch, any pre-training using data from another language leads to improved performance. Fine-tuning the model trained on English data yields the same average accuracy but with slightly improved stability. This phenomenon can be attributed to Dutch’s intermediary position between English and German. Consequently, it can leverage the generic difficulty information captured in the examples of both languages. Moreover, Dutch benefits from having the highest number of examples, allowing the second fine-tuning step to overturn biases related to specific aspects of the other languages. In contrast, German, arguably the most grammatically complex of the three languages and having the fewest examples, demonstrates improved performance only when fine-tuned with the model trained on English and German data. The limited contribution of data from other languages is insufficient to address German’s more complex grammatical aspects, making it challenging to overturn previous biases. Overall, the optimal performance for every language is achieved by fine-tuning the model trained on English and German data. This suggests that, due to the higher number of Dutch examples, the background models trained on Dutch data fail to appropriately attribute significance to specific aspects of the other languages.

In terms of adjacent accuracy, for English and Dutch, the patterns mirror those observed in terms of accuracy, with the highest performance differing by less than half a percentage point from that seen in the single-step training scenario. This minor variation was expected since the performance in both cases is close to perfect. In contrast, for German, additional models surpass the performance of the one trained solely on German data. Notably, the highest-performing model achieves over 90% adjacent accuracy and is trained on English and Dutch data in the first step. This finding further supports the claim that some cross-language aspects influence text readability assessment.

Finally, in Table 5, we observe that, akin to when considering the six CEFR levels, when only three levels are considered, the overall performance also improves compared to the single-step training scenario. Notably, for German, the highest performance is achieved by fine-tuning the model trained on English and German data in the first step, similarly to what was observed in the six-level scenario. However, for English, the highest performance is now attained by fine-tuning the model trained on English and Dutch data in the first step,

■ **Table 5** Accuracy (%) results using two-step training for three CEFR levels.

	English	Dutch	German
EN	79.11±1.92	85.28±0.64	80.07±2.26
NL	80.89±2.04	84.72±1.58	80.39±1.70
DE	80.22±3.79	84.58±1.67	81.04±5.74
EN+NL	81.55±2.04	85.97±0.64	81.05±3.00
EN+DE	80.45±2.04	86.25±1.10	83.01±1.50
NL+DE	80.00±2.40	85.00±1.25	78.76±3.44
All	80.22±4.29	85.56±1.68	81.37±1.70

which is consistent with the findings from the single-step training scenario. This observation suggests that, from a broader perspective, certain aspects contributing to text difficulty are comparable in both languages, and that the prediction for the English language benefits from the larger number of examples available in Dutch. Once again, the adjacent accuracy results are omitted because every model achieves 100% performance, rendering the metric uninformative in this context.

4.3 Overall Discussion

Based on the results presented in the preceding sections, it is evident that models trained in one language cannot be directly applied to predict the readability level of texts in other languages. Nevertheless, data from diverse languages can be utilized to create background models that are able to capture certain cross-language aspects of text readability, which can subsequently be adapted to specific languages. However, indiscriminately adding training data from different languages does not appear to be the most effective approach, though clear patterns of influence between related languages were not identified. Therefore, crafting specialized models for each language remains essential to achieve optimal performance on the task.

Nonetheless, this line of research presents several promising avenues to pursue, which can yield more robust conclusions and help address the identified issues. Specifically, the datasets used in this study are limited in both size and the number of languages considered. Expanding the data to encompass additional sources could potentially lead to the development of a more generic model of text readability, serving as a universal starting point for training language-specific models. Moreover, the choice of the base model, particularly the data used for its pre-training, may contribute to imbalances in performance across different languages. Therefore, it is crucial to explore alternative base models, especially considering the recent advancements in the field of large language models. Taking a more language-specific approach, involving proficient speakers of each language to examine the prediction errors of the models becomes essential. Such analysis might reveal specific patterns that the current models miss and highlight potential discrepancies in annotation across languages.

To conclude this discussion, a remark concerning variability deserves attention. Despite the apparent high variability observed for some models, it is essential to acknowledge that (except for some low-performance models) this variability stems from the differing difficulty levels of the test sets in each split. Most importantly, every claim made in the results' discussion is not solely based on average results but also on the consistency observed across multiple runs.

5 Conclusion

In this study, we conducted an investigation into automatic readability level classification from a cross-language perspective, focusing on the CEFR levels. Our primary objectives were twofold: first, to evaluate the potential of data from one language to enhance classification performance in different languages, and, second, to explore the effectiveness of utilizing scarce labeled resources more efficiently. To achieve these goals, we employed a pre-trained multilingual Transformer-based language model, fine-tuned it using annotated data from one language or a combination of languages, and then examined its generalization capacity even to unseen languages. Additionally, we performed further fine-tuning of the models using data from the target language to assess whether models trained on data in different languages were able to capture generic information related to text readability and could subsequently be further specialized to capture the specific characteristics of the target language. Through experiments covering the English, Dutch, and German languages, we arrived at insightful conclusions. It became apparent that direct generalization to unseen languages presents considerable challenges. However, when paired with data from the target language, multilingual data can be leveraged and proves beneficial to the task, as it allows the models to capture cross-language aspects of text readability. This leads to the development of more robust and improved-performing models overall.

In conclusion, our study has yielded compelling findings that open up a new avenue for training automatic readability assessment models. However, several limitations warrant consideration for future research. Chiefly, the datasets used in the study are constrained in size and encompass only a limited number of languages of the same family. Consequently, further investigations should evaluate the potential of incorporating additional data to establish more robust conclusions concerning cross-language interactions and to develop a more generic model of text readability, which can subsequently be fine-tuned for specific languages. For instance, it would be interesting to explore the 13 languages currently covered by the UNIVERSALCEFR initiative and assess whether the patterns observed in this study are replicated in other families, as well as in cross-family settings.

While our study prioritized exploring the cross-language aspects of automatic readability level classification rather than aiming for the highest performance possible, the findings lay the groundwork for future research, opening for comparison with higher-performing base models. It is essential to acknowledge that multilingual models may exhibit reduced performance compared to their single-language counterparts. Consequently, a critical consideration is whether the performance gains achieved by leveraging data from different languages can sufficiently compensate for the lower base performance of multilingual models. A potential path to address this concern is to leverage data from different languages by translating it into the target language instead of training a multilingual model directly on it. However, text complexity aspects can be lost or increased in translation. As such, future investigations should take these factors into account for a more comprehensive assessment of model performance and a more accurate estimation of the impact of multilingual data on the task of readability level classification.

Another important aspect to consider is that the current top performing approaches to text readability assessment rely on hybrid models that combine model fine-tuning with the use of handcrafted features. While this has not been thoroughly explored, the importance of specific features for readability assessment is expected to show higher variability across languages. Thus, the use of multilingual data in the context of hybrid models poses additional challenges that require further research.

References

- 1 Wafa Aissa, Raquel Amaro, David Antunes, Thibault Bañeras-Roux, Jorge Baptista, Alejandro Catala, Luís Correia, Thomas François, Marcos Garcia, Mario Izquierdo-Álvarez, Nuno Mamede, Vasco Martins, Miguel Neves, Eugénio Ribeiro, Sandra Rodriguez, and Elodie Vanzeveren. The iRead4Skills Intelligent Complexity Analyzer. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 73–84, 2025. doi:10.18653/v1/2025.emnlp-demos.6.
- 2 Wafa Aissa, Thibault Bañeras-Roux, Elodie Vanzeveren, Lingyun Gao, Rodrigo Wilkens, and Thomas François. Assessing French Readability for Adults with Low Literacy: A Global and Local Perspective. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 20528–20550, 2025. doi:10.18653/v1/2025.emnlp-main.1036.
- 3 Soroosh Akef, Amália Mendes, Detmar Meurers, and Patrick Rebuschat. Investigating the Generalizability of Portuguese Readability Assessment Models Trained Using Linguistic Complexity Features. In *Proceedings of the International Conference on Computational Processing of Portuguese (PROPOR)*, pages 332–341, 2024. URL: <https://aclanthology.org/2024.propor-1.34/>.
- 4 Sandra Aluisio, Lucia Specia, Caroline Gasperin, and Carolina Scarton. Readability Assessment for Text Simplification. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9, 2010. URL: <https://aclanthology.org/W10-1001>.
- 5 António Branco, João Rodrigues, Francisco Costa, João Silva, and Rui Vaz. Rolling out Text Categorization for Language Learning Assessment Supported by Language Technology. In *Proceedings of the International Conference on the Computational Processing of the Portuguese Language (PROPOR)*, pages 256–261, 2014. doi:10.1007/978-3-319-09761-9_29.
- 6 Mark Breuker. CEFR Labelling and Assessment Services. In Georg Rehm, editor, *European Language Grid: A Language Technology Platform for Multilingual Europe*, pages 277–282. Springer International Publishing, 2023. doi:10.1007/978-3-031-17258-8_16.
- 7 Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.
- 8 Miriam Cha, Youngjune Gwon, and HT Kung. Language Modeling by Clustering with Word Embeddings for Text Readability Assessment. In *Proceedings of the Conference on Information and Knowledge Management (CIKM)*, pages 2003–2006, 2017. doi:10.1145/3132847.3133104.
- 9 Council of Europe. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Cambridge University Press, 2001. URL: <https://rm.coe.int/1680459f97>.
- 10 Scott A. Crossley, Stephen Skalicky, Mihai Dascalu, Danielle S. McNamara, and Kristopher Kyle. Predicting Text Comprehension, Processing, and Familiarity in Adult Readers: New Approaches to Readability Formulas. *Discourse Processes*, 54(5-6):340–359, 2017. doi:10.1080/0163853x.2017.1296264.
- 11 Pedro Curto, Nuno Mamede, and Jorge Baptista. Automatic Text Difficulty Classifier. In *Proceedings of the International Conference on Computer Supported Education (CSEDU)*, volume 1, pages 36–44, 2015. doi:10.5220/0005428300360044.
- 12 Jacob Devlin, Ming-Wei Chang, Lee Kenton, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, volume 1, pages 4171–4186, 2019. doi:10.18653/v1/N19-1423.

- 13 William H. DuBay. *The Principles of Readability*. Impact Information, 2004. URL: <https://files.eric.ed.gov/fulltext/ED490073.pdf>.
- 14 Anna Filighera, Tim Steuer, and Christoph Rensing. Automatic Text Difficulty Estimation Using Embeddings and Neural Networks. In *Proceedings of the European Conference on Technology Enhanced Learning (EC-TEL)*, pages 335–348, 2019. doi:10.1007/978-3-030-29736-7_25.
- 15 Luciana Forti, Giuliana Grego Bolli, Filippo Santarelli, Valentino Santucci, and Stefania Spina. MALT-IT2: A New Resource to Measure Text Difficulty in Light of CEFR Levels for Italian L2 Learning. In *Proceedings of Language Resources and Evaluation Conference (LREC)*, pages 7204–7211, 2020. URL: <https://aclanthology.org/2020.lrec-1.890>.
- 16 Thomas François and Cédric Fairon. An “AI readability” Formula for French as a Foreign Language. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 466–477, 2012. URL: <https://aclanthology.org/D12-1043>.
- 17 Thomas François, Adeline Müller, Eva Rolin, and Magali Norré. AMesure: A Web Platform to Assist the Clear Writing of Administrative Texts. In *Proceedings of the Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP): System Demonstrations*, pages 1–7, 2020. doi:10.18653/V1/2020.AACL-DEMO.1.
- 18 Gemma Team. Gemma: Open Models based on Gemini Research and Technology. *Computing Research Repository*, arXiv:2403.08295, 2024. doi:10.48550/arXiv.2403.08295.
- 19 Nicolas Hernandez, Nabil Oulbaz, and Tristan Faine. Open Corpora and Toolkit for Assessing Text Readability in French. In *Proceedings of the Workshop on Tools and Resources to Empower People with READING Difficulties (READI)*, pages 54–61, 2022. URL: <https://aclanthology.org/2022.readi-1.8>.
- 20 Joseph Marvin Imperial, Abdullah Barayan, Regina Stodden, Rodrigo Wilkens, Ricardo Muñoz Sánchez, Lingyun Gao, Melissa Torgbi, Dawn Knight, Gail Forey, Reka R. Jablonkai, Ekaterina Kochmar, Robert Joshua Reynolds, Eugénio Ribeiro, Horacio Saggion, Elena Volodina, Sowmya Vajjala, Thomas François, Fernando Alva-Manchego, and Harish Tayyar Madabushi. UniversalCEFR: Enabling Open Multilingual Research on Language Proficiency Assessment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9703–9755, 2025. doi:10.18653/v1/2025.emnlp-main.491.
- 21 Simon Jönsson, Evelina Rennes, Johan Falkenjack, and Arne Jönsson. A Component Based Approach to Measuring Text Complexity. In *Proceedings of the Swedish Language Technology Conference (SLTC)*, pages 58–61, 2018. URL: <https://sltc2018.blogs.dsv.su.se/files/2019/11/Final-compilation-of-abstracts-SLTC-2018-Nov-19.pdf>.
- 22 Nikolay Karpov, Julia Baranova, and Fedor Vitugin. Single-sentence Readability Prediction in Russian. In *Proceedings of the International Conference on Analysis of Images, Social Networks and Texts (AIST)*, pages 91–100, 2014. doi:10.1007/978-3-319-12580-0_9.
- 23 J. Peter Kincaid, Robert P. Fishburne Jr, Richard L. Rogers, and Brad S. Chissom. Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel. Technical report, Institute for Simulation and Training, University of Central Florida, 1975. doi:10.21236/ada006655.
- 24 Sidney Evaldo Leal, Magali Sanches Duran, Carolina Evaristo Scarton, Nathan Siegle Hartmann, and Sandra Maria Aluísio. NILC-Matrix: Assessing the Complexity of Written and Spoken Language in Brazilian Portuguese. *Computing Research Repository*, arXiv:2201.03445, 2021. doi:10.48550/arXiv.2201.03445.
- 25 Bruce W. Lee, Yoo Sung Jang, and Jason Lee. Pushing on Text Readability Assessment: A Transformer Meets Handcrafted Linguistic Features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10669–10686, 2021. doi:10.18653/v1/2021.emnlp-main.834.

- 26 Fengkai Liu and John SY Lee. Hybrid models for sentence readability assessment. In *Proceedings of the Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 448–454, 2023. doi:10.18653/v1/2023.bea-1.37.
- 27 Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *Computing Research Repository*, arXiv:1907.11692, 2019. doi:10.48550/arXiv.1907.11692.
- 28 Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. Supervised and Unsupervised Neural Approaches to Text Readability. *Computational Linguistics*, 47(1):141–179, 2021. doi:10.1162/coli_a_00398.
- 29 Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G.C. de Souza, Alexandra Birch, and André F.T. Martins. EuroLLM: Multilingual Language Models for Europe. *Procedia Computer Science*, 255:53–62, 2025. doi:10.1016/j.procs.2025.02.260.
- 30 Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *NeurIPS*, pages 3111–3119, 2013. URL: <https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality>.
- 31 Salar Mohtaj, Babak Naderi, and Sebastian Möller. Overview of the GermEval 2022 Shared Task on Text Complexity Assessment of German Text. In *Proceedings of the GermEval Workshop on Text Complexity Assessment of German Text*, pages 1–9, 2022. URL: <https://aclanthology.org/2022.germEval-1.1>.
- 32 Farah Nadeem and Mari Ostendorf. Estimating Linguistic Complexity for Science Texts. In *Proceedings of the Workshop on Innovative Use of NLP for Building Educational Applications*, pages 45–55, 2018. doi:10.18653/v1/W18-0505.
- 33 Kai North, Marcos Zampieri, and Matthew Shardlow. Lexical Complexity Prediction: An Overview. *ACM Computing Surveys*, 55(9):1–42, 2023. doi:10.1145/3557885.
- 34 Ildikó Pilán and Elena Volodina. Investigating the Importance of Linguistic Complexity Features Across Different Datasets Related to Language Learning. In *Proceedings of the Workshop on Linguistic Complexity and Natural Language Processing*, pages 49–58, 2018. URL: <https://aclanthology.org/W18-4606>.
- 35 Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language Models are Unsupervised Multitask Learners. *OpenAI Blog*, 1(8):9, 2019. URL: https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.
- 36 Sandra Rodríguez Rey, André Bernárdez Braña, and Marcos Garcia. Exploring Linguistic Features in a New Readability Corpus for Spanish. *Procesamiento del lenguaje natural*, 74:221–239, 2025. URL: <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6677>.
- 37 Sandra Rodríguez Rey and Marcos Garcia. Clasificación Automática de Textos por Níveis de Lecturabilidade: Recursos e Modelos para o Galego. *Linguamática*, 17(2):33–56, 2025. doi:10.21814/lm.17.2.488.
- 38 Robert Reynolds. Insights from Russian Second Language Readability Classification: Complexity-Dependent Training Requirements, and Feature Evaluation of Multiple Categories. In *Proceedings of the Workshop on Innovative Use of NLP for Building Educational Applications*, pages 289–300, 2016. doi:10.18653/v1/W16-0534.
- 39 Eugénio Ribeiro, David Antunes, Nuno Mamede, and Jorge Baptista. Exploring Few-Shot Approaches to Automatic Text Complexity Assessment in European Portuguese. *Journal of the Brazilian Computer Society*, 31(1):690–710, 2025. doi:10.5753/jbcs.2025.5820.
- 40 Eugénio Ribeiro, Nuno Mamede, and Jorge Baptista. Automatic Text Readability Assessment in European Portuguese. In *Proceedings of the International Conference on Computational Processing of Portuguese (PROPOR)*, pages 97–107, 2024. URL: <https://aclanthology.org/2024.propor-1.10/>.

- 41 Eugénio Ribeiro, Nuno Mamede, and Jorge Baptista. Avaliação Automática do Nível de Complexidade de Textos em Português Europeu. *Linguamática*, 16(2):121–145, 2024. doi: 10.21814/lm.16.2.449.
- 42 Eugénio Ribeiro, Nuno Mamede, and Jorge Baptista. Text Readability Assessment in European Portuguese: A Comparison of Classification and Regression Approaches. In *Proceedings of the International Conference on Computational Processing of Portuguese (PROPOR)*, pages 551–557, 2024. URL: <https://aclanthology.org/2024.propor-1.59/>.
- 43 Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *Computing Research Repository*, arXiv:1910.01108, 2019. doi:10.48550/arXiv.1910.01108.
- 44 Rodrigo Santos, João Rodrigues, António Branco, and Rui Vaz. Neural Text Categorization with Transformers for Learning Portuguese as a Second Language. In *Proceedings of the Portuguese Conference on Artificial Intelligence (EPIA)*, pages 715–726, 2021. doi:10.1007/978-3-030-86230-5_56.
- 45 Valentino Santucci, Filippo Santarelli, Luciana Forti, and Stefania Spina. Automatic Classification of Text Complexity. *Applied Sciences*, 10(20):7285, 2020. doi:10.3390/app10207285.
- 46 Yao Ting Sung, Wei Chun Lin, Scott Benjamin Dyson, Kuo En Chang, and Yu Chia Chen. Leveling L2 Texts through Readability: Combining Multilevel Linguistic Features with the CEFR. *The Modern Language Journal*, 99(2):371–391, 2015. doi:10.1111/modl.12213.
- 47 Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models. *Computing Research Repository*, arXiv:2302.13971, 2023. doi:10.48550/arXiv.2302.13971.
- 48 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, pages 5998–6008, 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
- 49 Rodrigo Wilkens, David Alfter, Xiaoou Wang, Alice Pintard, Anaïs Tack, Kevin Yancey, and Thomas François. FABRA: French Aggregator-Based Readability Assessment Toolkit. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 1217–1233, 2022. URL: <https://aclanthology.org/2022.lrec-1.130>.
- 50 Rodrigo Wilkens, Patrick Watrin, Rémi Cardon, Alice Pintard, Isabelle Gribomont, and Thomas François. Exploring Hybrid Approaches to Readability: Experiments on the Complementarity between Linguistic Features and Transformers. In *Findings of the Association for Computational Linguistics: EACL*, pages 2316–2331, 2024. doi:10.18653/v1/2024.findings-eacl.153.
- 51 Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace’s Transformers: State-of-the-art Natural Language Processing. *Computing Research Repository*, arXiv:1910.03771, 2019. doi:10.48550/arXiv.1910.03771.
- 52 Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. Text Readability Assessment for Second Language Learners. In *Proceedings of the Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22, 2016. doi:10.18653/v1/W16-0502.
- 53 Kevin Yancey, Alice Pintard, and Thomas Francois. Investigating Readability of French as a Foreign Language with Deep Learning and Cognitive and Pedagogical Features. *Lingue e Linguaggio*, 20(2):229–258, 2021. doi:10.1418/102814.

BERT-Based Classification of Cross-Linguistic MCQs According to the Bloom Taxonomy

João Francisco Botas¹  

Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, Lisboa, Portugal

Ana Rita Peixoto  

Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, Lisboa, Portugal

Eugénio Ribeiro  

INESC-ID Lisboa, Portugal

Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, Lisboa, Portugal

Abstract

The Bloom Taxonomy provides a useful framework for describing the cognitive demands of Multiple Choice Questions (MCQs), yet automatically assigning Bloom levels remains challenging due to category overlap and the limited evidence of higher-order thinking in MCQs. This study aims to evaluate transformer-based models, including BERT and ModernBERT variants, to enhance their performance on a four-level Bloom classification task across multiple scenarios, while also conducting model interpretability and error analysis.

Across 40 settings tested, BERT models consistently outperform a keyword-based baseline, highlighting the contextual and semantic representations for this task. Performance is broadly similar between BERT and ModernBERT, with only minor differences across architectures, while the inclusion of answer options yields only small gains, suggesting that the question stem alone contains most of the discriminative signal. Besides, cross-linguistic experiments show comparable results between the original English data and Portuguese, although translated data exhibits a slight performance degradation, likely due to direct translation noise and subtle syntactic shifts. Errors are concentrated between adjacent Bloom levels, and models perform best on the *Remembering* level. In contrast, the *Applying* level remains the most difficult class to predict, largely because of class imbalance and conceptual overlap.

The findings indicate that our approach constitutes a stable pipeline for Bloom classification, but its performance remains constrained by the intrinsic ambiguity of the taxonomy and the structural characteristics of the available data.

2012 ACM Subject Classification Computing methodologies → Natural language processing

Keywords and phrases Bloom Taxonomy, Multiple Choice Questions, Transformers, Multi-Language, Natural Language Processing, Educational Assessment

Digital Object Identifier 10.4230/OASICS.SLATE.2026.4

Funding This work was partially supported by national funds through Fundação para a Ciência e a Tecnologia, I.P. (FCT), ISTAR-Iscte Pluriannual Projects UID/04466/2025 (<https://doi.org/10.54499/UID/04466/2025>) and INESC-ID Projects UID/50021/2025 (<https://doi.org/10.54499/UID/50021/2025>) and UID/PRR/50021/2025 (<https://doi.org/10.54499/UID/PRR/50021/2025>).

1 Introduction

Assessing the difficulty and complexity of educational questions remains a challenging task, particularly in large-scale settings where Multiple Choice Question (MCQ) are widely used. Traditional approaches rely on pretesting procedures, pilot studies, or expert judgment

¹ Corresponding author



© João Francisco Botas, Ana Rita Peixoto, and Eugénio Ribeiro;
licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 4; pp. 4:1–4:17

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

to assign difficulty levels, but these methods are costly, time-consuming, and difficult to scale [11]. As a result, there is a growing interest in automated approaches that can infer question characteristics directly from textual and structural features, enabling more efficient analysis of large educational datasets.

A key dimension in understanding question complexity is not only in “how difficult a question is”, but also “what type of cognitive process it requires to be answered”. In this context, Bloom’s Taxonomy (BT) provides a structured framework to characterize the level of thinking involved in answering a question, organizing cognitive skills into six hierarchical levels: 1-*Remembering*, 2-*Understanding*, 3-*Applying*, 4-*Analyzing*, 5-*Evaluating*, and 6-*Creating* (for the Revised Bloom’s Taxonomy (BT)) [3]. Unlike traditional difficulty measures, which are often derived from exhaustive student performance data, BT focuses on the item’s intended cognitive demand, making it suitable for analyzing question design [24]. However, applying this framework in practice is non-trivial due to the cognitive levels often overlapping, being context-dependent, and being subjective to interpret, including for specific types of questions, such as MCQs [22]. Despite these challenges, BT offers a meaningful proxy for complexity that can be operationalized through multiple Natural Language Processing (NLP) techniques, from the more traditional to the most recent, enabling automated analysis of educational content [18].

In this study, we investigate the use of transformer-based models, specifically BERT architectures [6], to automatically classify MCQs according to BT levels. Our approach focuses on evaluating how well these models capture the semantic and contextual information required to distinguish between cognitive levels, using both the question text and, optionally, the answer options as input. Furthermore, we extend this analysis to a multilingual setting by comparing model performance across English and Portuguese versions of the same dataset, using specific BERT models for each. This allows us to examine the impact of language variation on model behavior, particularly when data is obtained through automatic translation. In addition to standard classification metrics, we also consider evaluation strategies that account for the ordinal nature of Bloom levels, providing a more nuanced understanding of model performance. We also try to analyze how different question formats (fill-in-blank versus non-fill-in-blank MCQs) and domain can affect classification performance and model global behavior. All of these aspects are systematically evaluated across multiple experimental configurations, enabling a comprehensive comparison of modeling choices and their impact on performance.

To guide this study, we define a set of Research Questions (RQs) organized around three main topics: model effectiveness, interlingual robustness, and error analysis. These research questions are designed to assess not only the overall performance of the proposed approaches but also to better understand their behavior under different configurations, languages, and data characteristics. The defined RQs are as follows:

1. To what extent can BERT-based models accurately classify MCQs according to Bloom’s Taxonomy (BT), and how do different configurations affect performance?
2. Do models trained on translated datasets maintain comparable performance across languages, and how do linguistic and syntactic variations influence classification outcomes?
3. What are the most common misclassification patterns, particularly between adjacent cognitive levels, and how are these affected by other data typologies?

The remainder of this document is organized as follows: Section 2 reviews related work on Bloom’s Taxonomy (BT) classification, MCQ assessment, and NLP approaches. Section 3 describes the datasets, preprocessing steps, model configurations, experimental

setup, and evaluation metrics. Section 4 presents the results, starting with the baseline and then analyzing the transformer-based models, including cross-linguistic comparisons, error patterns, and profiling analyses. Finally, Section 5 summarizes the main findings, limitations, and future work.

2 Related Work

Bloom’s Taxonomy (BT) has been widely adopted as a framework to classify cognitive processes in educational assessment, and organizing skills in a hierarchical structure where higher-order abilities depend on the mastery of lower-order ones [14, 15]. Despite its theoretical clarity, the practical application of BT remains challenging, as educators often struggle to consistently define and distinguish between cognitive levels due to their progressive dependencies and overlapping competencies [14]. The classification of questions is also inherently subjective since learners may interpret the same item differently depending on their knowledge, confidence, and problem-solving strategies [18, 29].

These challenges are particularly evident in the context of MCQs, one of the most widely used assessment formats in education [16, 23]. While MCQs are effective in evaluating factual knowledge (*Remembering*) and even higher cognitive skills, research consistently shows that they predominantly target the lowest levels of BT [23]. Although well-designed MCQs may assess intermediate levels such as *Applying* and *Analyzing*, the highest cognitive processes, of *Evaluation* and *Creation*, are rarely captured, as they typically require open-ended responses [15]. Moreover, even within the same question type of MCQs, items can vary substantially in complexity, a distinction that broad taxonomic levels can fail to capture [22]. To address these limitations, several works adopt a simplified representation of BT by grouping levels into lower-order and higher-order categories, thereby improving results and better reflecting the practical constraints of MCQ-based assessment [19, 31].

Classification using Bloom’s Taxonomy

Early NLP approaches for BT classification relied on keyword/rule-based methods, using simple pre-processing techniques and predefined verb lists, or n-gram classifiers [8]. While achieving reasonable performance on lower levels, such as *Remembering* (around 75% accuracy), they perform poorly on higher levels, reflecting their inability to capture deeper semantic meaning [4, 20]. Building upon these limitations, more advanced approaches have explored emerging Machine Learning (ML) and NLP techniques to better capture semantic and contextual information.

In non-MCQ settings, transformer-based models such as BERT have demonstrated superior performance compared to traditional methods, significantly improving classification metrics across Bloom levels and effectively modeling contextual dependencies in educational texts [14, 21]. Additionally, studies in other languages and domains, such as Indonesian question classification, have combined lexical and syntactic feature extraction with ML models, including SVMs. These studies demonstrate that BT classification can generalize beyond English, although performance remains dependent on feature quality, POS-tagging-based feature extraction, and dataset size [13, 14]. These findings suggest that, while BT captures cognitive processes that are largely language independent, its classification remains inherently complex across linguistic settings. To address possible language-related limitations, other NLP studies, involving sentiment analysis or semantic similarity, have explored translation strategies, often achieving consistent or even improved performance over the native language [26].

Focusing on MCQ-based contexts, which introduce additional structural complexity, based on shorter question lengths, recent works have focused on leveraging transformer architectures with more sophisticated input representations to model relationships between question stems and answer options (e.g., using separated inputs such as “[SEP] Options” tokens) [24]. BERT models have been applied to classify MCQs according to BT, showing good performance, particularly at lower cognitive levels. Reported results indicate that classification accuracy improves significantly when simplifying the label space, increasing from 59.2% in the full six-level setting to 68.52% when merging underrepresented categories, and reaching 82.61% when further removing these classes [32]. These improvements are largely driven by the aggregation of less frequent classes, particularly the *Applying*, which is merged with higher-order categories to mitigate class imbalance and improve model learning. Similar trends are observed in Naive Bayes classifiers, where collapsing all of the BT levels, except *Remembering*, also leads to improved performance, albeit with limitations in cross-domain generalization [19]. Overall, these approaches are constrained by the imbalance and relatively small size of available datasets, which hinder the learning of higher-order cognitive distinctions [5]. Although performance improvements of the model understanding often come at the cost of reduced label granularity, as simplifying higher-order levels limits the ability to fully represent BT cognitive hierarchy [19, 32].

Even more recent approaches explore Large Language Models (LLMs) and specialized architectures for MCQ analysis, achieving strong performance on factual tasks, but still struggle with higher-order reasoning, with performance decreasing with cognitive complexity and item difficulty [2]. While these methods are not directly focused on BT classification, but rather on their generation, they provide relevant insights into how models handle different cognitive levels. In particular, these LLMs can generate and classify Bloom questions, but ensuring consistent quality and alignment across cognitive levels remains challenging, with discrepancies observed between automated and human evaluation [12]. Furthermore, as mentioned previously, the MCQ format itself introduces additional limitations to these strategies, as its structure can obscure item discrimination and hinder the reliable assessment of higher-order cognitive processes [9, 23].

3 Methodology

This section describes the study’s pipeline, including the datasets, experimental setup, hyperparameter configurations, and evaluation metrics for the proposed models.

3.1 Data

In this study, we utilized two publicly available educational datasets, namely EduQG² and EduQuest³ [7, 10]. From the latter, we primarily leveraged questions sourced from the OpenStax platform, covering Economics (3e), Microeconomics (2e and 3e), and Organizational Behavior. In contrast, EduQG does not provide explicit domain labels, but mainly contains science-related questions. Overall, merging these sources results in a more diverse dataset, both in terms of subject coverage and question structure.

To prepare these datasets, we first filtered only MCQs, which were identified either through an existing boolean indicator provided in the datasets or programmatically by parsing Question+Options structures in the text column. Then, only instances annotated

² <https://github.com/EchizenKurage/EduQG>

³ <https://github.com/aegonwolf/EduQuest>

■ **Table 1** Processed dataset summary statistics for each Bloom cognitive level (the last 3 columns represent a mean and standard deviation for each Bloom level).

BT Level	Count (%)	OPTs < 3 words	Fillblank ratio	Question char	OPTs char	OPTs words
1 – Remembering	1132 (26.88%)	516	49.29%	92 ± 51	103 ± 113	15 ± 17
2 – Understanding	1737 (41.25%)	372	45.83%	119 ± 70	156 ± 127	23 ± 20
3 – Applying	335 (7.96%)	147	27.76%	206 ± 167	122 ± 147	19 ± 24
4 – Analyzing	1007 (23.91%)	258	24.73%	214 ± 172	156 ± 119	25 ± 19

with a BT cognitive level were considered, ranging originally from 1 (*Remembering*) to 6 (*Creating*). In addition, to ensure greater consistency and mitigate class imbalance, only the first four levels of BT were considered. This decision was motivated mainly by: i) the limited representation of these classes in the merged dataset (166 and 16 instances out of a total of 4393 for *Evaluating* and *Creating* levels, respectively), and ii) the difficulty of identifying the top two levels of high-order thinking within the context of MCQs [15].

The merged dataset differs in both structure and domain coverage, encompassing variations in subject matter, as well as in question formats, including both standard and “fill-in-blank” MCQ items. This heterogeneity introduces increased complexity to the task while also supporting the research objective of evaluating the model’s global performance in a cross-domain setting. Table 1 summarizes the main characteristics of the resulting dataset across the four BT levels considered in this study.

As shown in Table 1, the distribution of instances is skewed toward lower cognitive levels (*Remembering* and *Understanding*), which together comprise the majority of the dataset, while *Applying* is notably underrepresented. This discrepancy may stem from the annotator’s subjective association of questions with BT levels, which can be inherently debatable. Despite the imbalance in sample sizes, some trends emerge, particularly structural differences in answer options. Lower levels, especially *Remembering*, show more options with fewer than three words, indicating a reliance on short, direct answers. However, this can be domain-dependent, as in questions involving numerical or formula-based reasoning, options often remain concise regardless of level, typically representing final results, which can bias length averages. Additionally, *Understanding* and *Analyzing* exhibit similar option lengths, suggesting that length alone does not capture cognitive complexity, while question length generally increases with higher levels.

The ratio of fill-in-blank items also decreases progressively with increasing cognitive level, indicating that higher-level questions rely less on simple recall formats.

3.2 Experimental setup

This section describes the experimental setup, including model configurations, input representations, and training procedures for the level-specific Bloom classification task.

As an initial approach, we implemented a solution based on predefined keyword/verb lists associated with each BT level (Table 2). Each question is assigned to a cognitive level based on the presence of specific keywords in the text. However, some action verbs, such as “classify”, may appear at multiple levels, introducing ambiguity into this method, which requires certain heuristics. Prior to keyword matching, we applied simple text preprocessing to both the BT keyword lists and the question texts, including lowercasing, stopword removal, and stemming, in order to reduce lexical variability and improve matching consistency.

■ **Table 2** BT levels and typical keywords of each Bloom level (adapted from [3]).

Bloom #	Bloom Level	Typical Keywords
1	Remembering	choose, define, find, how, label, list, match, name, omit, recall, relate, select, show, spell, tell, what, when, where, which, who, why
2	Understanding	classify, compare, contrast, demonstrate, explain, extend, illustrate, infer, interpret, outline, relate, rephrase, show, summarize, translate, understand
3	Applying	apply, build, choose, construct, develop, experiment, with, identify, interview, make, use, of, model, organize, plan, select, solve, utilize
4	Analyzing	analyze, assume, categorize, classify, compare, conclusion, contrast, discover, dissect, distinguish, divide, examine, function, inference, inspect, list, motive, relationships, simplify, survey, take, part, in, test, for, theme

However, this approach has several limitations, as it is highly sensitive to ambiguity, with the same keyword often corresponding to multiple cognitive levels; questions may contain a variety of relevant keywords or none at all, making the decision less straightforward. In such cases, tie-breaking rules are required, introducing heuristics that may not generalize well. Despite these limitations, this method provides a simple and interpretable baseline for comparison with the fine-tuned BERT-based models, although its performance is expected to be very limited.

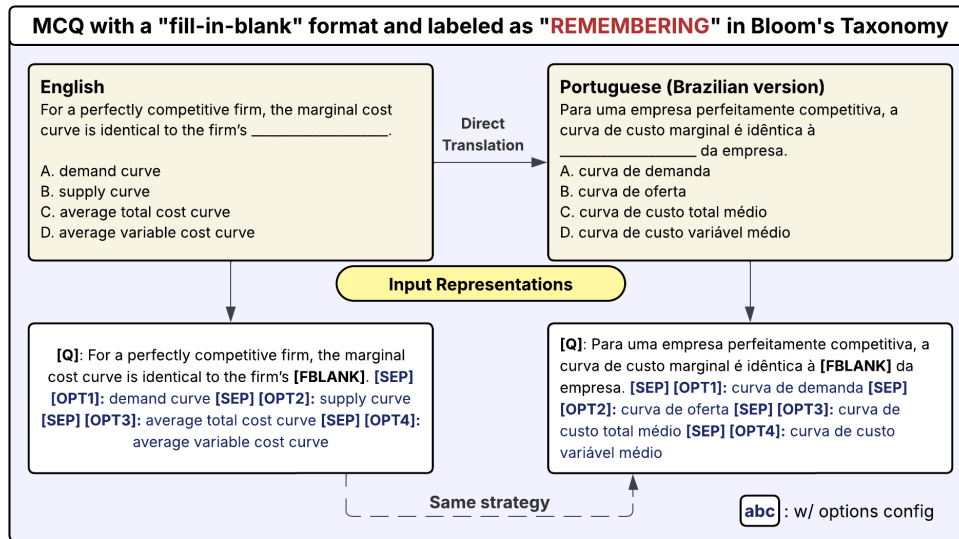
Building upon the baseline approach, we explore transformer-based models to capture contextual and semantic information, with the main objective not only to evaluate global performance but also to assess the model’s ability to distinguish between consecutive cognitive levels and to analyze disparities across languages. In particular, we aim to investigate whether models trained on the same data but directly translated into Brazilian Portuguese exhibit significant differences compared to their original form (English).

To this end, we employed two variants of pre-trained models for each language: i) a BERT-base version, and ii) a ModernBERT version [30]. For English, a standard BERT-base model and a ModernBERT model were used, each with 110M and 149M parameters, respectively. For Portuguese, we used the base version of BERTimbau (110M parameters) and a multilingual version of ModernBERT to assess the language [28, 17]. For comparison, the smaller multilingual ModernBERT model was chosen to ensure a similar number of parameters (140M). Regarding these Portuguese configurations, we obtained the translated version of the dataset using the **deep-translator** package⁴.

To ensure comparability across these four models running process, the same data split is used for both versions, with an 80/20 train–test ratio and stratification over the target labels. A fixed random seed is applied so that predictions on the test set remain coherent across runs of these different models.

The input representation for all models was unified and standardized across all experiments to reduce noise and redundant information. Each instance is encoded as a sequence combining

⁴ <https://pypi.org/project/deep-translator/>



■ **Figure 1** Example of input representations for the original English version and the Brazilian Portuguese translation.

the question and, optionally, its answer options, with the following structure: “[Q] <question text>” (with [FBLANK] tokens replacing blanks, when applicable), followed by “[SEP]” and the options starting as “[OPT1] <option 1 text>”. These representation and masking techniques are applied consistently to both English and Portuguese inputs, as illustrated in Figure 1, through a fill-in-blank *Remembering* MCQ example. Based on this formulation, we explicitly control the inclusion of answer options in the input sequence. Therefore, all configurations are evaluated under two conditions: with and without answer options, enabling a direct assessment of their impact on the model’s ability to infer the specific Bloom level.

BERT Hyperparameter tuning

Hyperparameter tuning was performed through a randomized search over predefined ranges for key parameters, including learning rate, weight decay, batch size, and warmup steps. To systematically explore this search space, we employed Optuna [1], running a total of 24 trials to identify the most suitable configurations. The optimization objective was the weighted F1-score, and the final hyperparameter settings were selected based on the 3 best-performing trials. So, the resulting configuration of hyperparameters used for all models in this study was essentially: **learning rate** = 1.25×10^{-5} , **training and evaluation batch size** = 16, **number of training epochs** = 10 (with a callback of early stopping using a patience of 2 epochs), **weight decay** = 0.04, and **warmup steps** = 0.1.

Class Selection

A complementary study was conducted to evaluate the impact of different class configurations on model performance. Initially, multiple setups were explored within a single model (BERT-base English), including merging adjacent levels (e.g., combining levels: {3, 4} or {3, 4, 5, 6}), removing higher levels ({3, 4} or {3, 5, 6}), or even both (e.g., dropping 5,6 and merging 3,4) [32, 19]. While some of these configurations yielded improved performance (primarily those

that reduced class granularity), the results are not directly comparable due to changes in the original classification task itself. Despite these improvements, the final decision was to maintain the four-level setting (dropping the 5-*Evaluation* and 6-*Creation* questions), as justified with the support of Table 1 and related work [19, 32]. Overall, evaluating each level independently was considered more meaningful for addressing RQ3 than merging classes, even at the cost of lower performance.

3.3 Evaluation metrics

To evaluate the performance of the fine-tuned models, we consider a set of standard classification metrics: accuracy, macro F1-score, weighted F1-score, and macro AUC. Accuracy provides a general measure of correctness but may be misleading in the presence of class imbalance. To address this, we report the macro F1-score, which treats all classes equally, highlighting performance on underrepresented classes, and the weighted F1-score that accounts for class frequency, in order to provide a more representative measure of overall performance. Given this class imbalance and to account for support, the weighted F1-Score is used as the primary metric for monitoring training and selecting the best model across epochs.

Beyond these traditional metrics, we also introduce additional evaluation measures, based on accuracy, to capture the ordinal nature of Bloom’s Taxonomy levels: Adjacent Accuracy and H/L Order Accuracy. Since misclassifications between adjacent levels are less severe than those across distant levels, these metrics provide a more nuanced and interpretable assessment of the traditional accuracy by accounting for levels within adjacent classes or subgroups.

Adjacent Accuracy ($\text{Adj}_{\pm 1}$) considers a prediction correct if it matches the true label or differs by at most one adjacent class level, as shown in Equation 1.

$$\text{Adj}_{\pm 1} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} (|\hat{y}_i - y_i| \leq 1) \quad (1)$$

In turn, H/L order accuracy evaluates whether predictions fall within the same ordinal group, being these groups: *Low-order* ($\{1, 2\}$) and *High-order* ($\{3, 4\}$). A prediction is considered correct if both predicted and true labels belong to the same group:

$$\text{H/L} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} (g(\hat{y}_i) = g(y_i)), \quad g(y) = \begin{cases} 0, & \text{if } y \in \{1, 2\} \text{ (Low)} \\ 1, & \text{if } y \in \{3, 4\} \text{ (High)} \end{cases} \quad (2)$$

This metric can be generalized to the full six-level Bloom taxonomy by redefining the grouping function accordingly. However, in our four-level setting, it behaves similarly to adjacent accuracy ($\text{Adj}_{\pm 1}$), effectively ignoring only confusions between the intermediate *Understanding* and *Applying* levels to focus on broader cognitive distinctions instead. In essence, this corresponds to a dichotomization ($\{1, 2\}$ and $\{3, 4\}$) of the resulting confusion matrix after training the model on the four-level classification task, rather than explicitly training a separate model for high/low-order classification; this would constitute a simpler task, as the model would not need to differentiate between all classes and would likely yield higher accuracy values in the binary classification than this new metric in the multi-class task.

Finally, in the evaluation phase, we explore two alternative strategies for interpreting the model output probabilities. The first corresponds to the standard *argmax* approach, where the predicted class is selected as the one with the highest probability, while the

second approach computes a *weighted average* of the class probabilities, leveraging the ordinal structure of the labels. This method, which led to more robust models in other NLP tasks [25], assigns greater weight to lower-probability classes, particularly at levels $\{3, 4\}$, which enables smoother predictions and provides additional insight into borderline cases where the model expresses uncertainty.

Besides, to reduce the impact of randomness during training, each experiment is repeated 5 times with different random seeds. The reported results correspond to the mean of the evaluation metrics across these runs, providing a more reliable estimate of model performance. Considering all configurations, this yields a total of 40 training runs, derived from 4 model variants (with the corresponding language), 2 input settings (with and without answer options), and 5 repetitions per configuration. The two methods shown to evaluate the probability vector don't affect the entire training (across the 40 training runs mentioned), but only the metrics and confusion matrix derived from the predictions.

4 Results

This section begins with an evaluation of the baseline model, keyword-based approach, followed by a comprehensive analysis of the BERT-based configurations. Multiple experimental setups are compared, with particular focus on their respective advantages and the identification of consistent performance patterns across models, input representations, and prediction strategies. Subsequently, a targeted error analysis is conducted to examine common misclassification trends, with special attention given to discrepancies observed between English and Portuguese predictions, supported by illustrative examples. Finally, the analysis is extended through the use of profiling variables, enabling a more granular assessment of model performance across Bloom levels, as well as across structural and contextual dimensions, such as fill-in-blank questions and topic-related categorical features.

Baseline Model

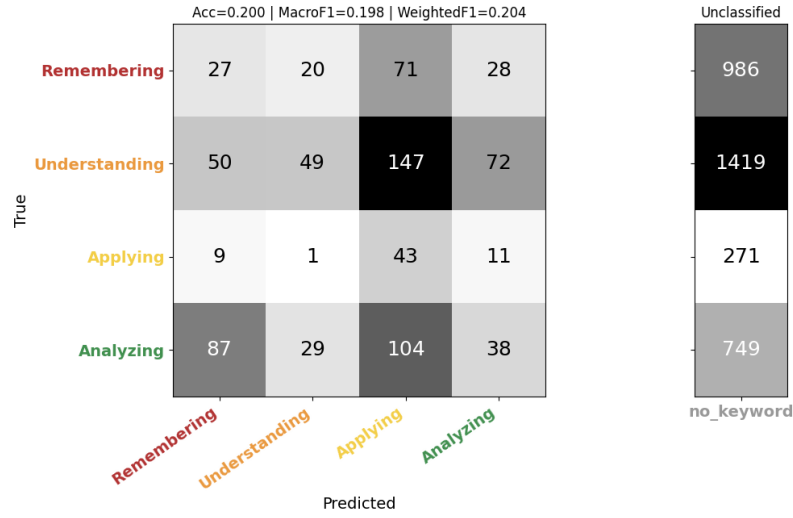
Before advancing to the evaluation of the learning-based models, it is important to analyze the performance of the baseline approach. First, it is relevant to notice that the baseline classifier operates deterministically over the entire dataset, as it does not involve any training phase, but instead assigns labels based on predefined keyword rules, already shown in Table 2. The resulting confusion matrix is shown in Figure 2.

The baseline classifier yields notably poor performance, as reflected in the low metric values (all close to 0.2). These results highlight the limitations of relying solely on keyword-based heuristics for Bloom level classification. From the confusion matrix, it is evident that the classifier exhibits a strong bias toward specific classes, with emphasis on *Applying* and, to a lesser extent, *Remembering*, which are frequently “overpredicted” regardless of the true label. For instance, a large portion of *Understanding* instances are incorrectly assigned to *Applying*, indicating that the presence of certain keywords may be misleading or overly generalized across levels.

Moreover, a substantial portion of the dataset remains unclassified, as many questions do not contain any of the predefined keywords. This is reflected in the large number of samples in the “no_keyword” category (+80% of the samples in $\{1, 2, 3\}$ levels and $\approx 74\%$ in the 4th level - *Analyzing*), which significantly limits the coverage and effectiveness of this approach.

These results reinforce the limitations of these simple rule-based methods, underscoring the need for models that capture deeper contextual and semantic information.

4:10 BERT-Based Classification of Cross-Linguistic MCQs According to the BT



■ **Figure 2** Confusion Matrix of the Baseline Model: Classification by keyword-based approach.

Transformer-based models

Moving on to the fine-tuned BERT models, Table 3 presents the performance of the different BERT-based configurations across languages, input settings, and prediction strategies. In this table, the colored arrows indicate the comparison by excluding the options in one color and changing the prediction strategy in another color. In turn, the arrows in ModernBERT compare each line to its mirror in the “simpler” BERT, without this clear color separation between options and strategy subgroups.

■ **Table 3** BERT-Models configurations results for the specific class of BT: mean after 5 runs of each config.

Model Configuration w/ the 1st 4 classes of BT				Classification Metrics (mean of 5 runs)					
Model	Lang.	+ OPT?	Pred. Strat.	Acc.	MacroF1	WeightedF1	Adj (± 1)	H/L order	Macro AUC
BERT-base	EN	✓	argmax	0.6989	0.6243	0.6922	0.8598	0.8273	0.8665
BERTimbau-base	PT	✓	argmax	0.6892	0.6201	0.6824	0.8486	0.8202	0.8635
BERT-base	EN	✗	argmax	0.6968 ↓	0.6209 ↓	0.6879 ↓	0.8520 ↓	0.8256 ↓	0.8625 ↓
BERTimbau-base	PT	✗	argmax	0.6878 ↓	0.6361 ↑	0.6837 ↓	0.8475 ↓	0.8187 ↓	0.8566 ↓
BERT-base	EN	✓	weighted avg	0.6486 ↓	0.5882 ↓	0.6613 ↓	0.8982 ↑	0.8244 ↓	0.8665
BERTimbau-base	PT	✓	weighted avg	0.6372 ↓	0.5754 ↓	0.6482 ↓	0.8887 ↑	0.8187 ↓	0.8635
BERT-base	EN	✗	weighted avg	0.6377 ↓↓	0.5780 ↓↓	0.6491 ↓↓	0.8963 ↓↑	0.8223 ↓↓	0.8625 ↓
BERTimbau-base	PT	✗	weighted avg	0.6489 ↑↓	0.5893 ↑↓	0.6552 ↑↓	0.8816 ↓↑	0.8211 ↑↑	0.8566 ↓
ModernBERT	EN	✓	argmax	0.6904 ↓	0.6292 ↑	0.6884 ↓	0.8550 ↓	0.8199 ↓	0.8686 ↑
mModern-small	PT	✓	argmax	0.6878 ↓	0.6224 ↑	0.6853 ↓	0.8503 ↑	0.8199 ↓	0.8590 ↓
ModernBERT	EN	✗	argmax	0.6871 ↓	0.6270 ↑	0.6838 ↓	0.8529 ↑	0.8230 ↓	0.8672 ↓
mModern-small	PT	✗	argmax	0.6840 ↓	0.6191 ↓	0.6793 ↓	0.8531 ↑	0.8202 ↑	0.8563 ↓
ModernBERT	EN	✓	weighted avg	0.6230 ↓	0.5718 ↓	0.6412 ↓	0.9006 ↑	0.8166 ↓	0.8686 ↑
mModern-small	PT	✓	weighted avg	0.6453 ↑	0.5889 ↑	0.6579 ↑	0.8859 ↓	0.8178 ↓	0.8590 ↓
ModernBERT	EN	✗	weighted avg	0.6380 ↑	0.5837 ↑	0.6502 ↑	0.8916 ↓	0.8214 ↓	0.8672 ↑
mModern-small	PT	✗	weighted avg	0.6313 ↓	0.5765 ↓	0.6441 ↓	0.8951 ↑	0.8192 ↓	0.8563 ↓

Overall, these transformer-based models outperform the baseline approach by a large margin, confirming the importance of contextual and semantic modeling for this task. This is also aligned with the nature of the annotation process itself, where human annotators rely

not only on lexical cues but also on semantic understanding, external knowledge, pattern recognition, and elements of critical thinking. While transformer models are capable of capturing contextual and semantic information, fully replicating the critical reasoning applied by human annotators remains a challenging aspect.

Across both BERT-base and ModernBERT variants, the results show very similar performance between the English and Portuguese models, in accuracy and weighted F1-score, above all. However, a slight and consistent performance drop is observed for the Portuguese versions. This can be attributed to the use of direct translation, which may introduce noise or subtle semantic inconsistencies, especially given the greater lexical variability of the Portuguese language. When comparing model architectures, ModernBERT exhibits behaviour similar to that of standard BERT-base models, although it does not consistently improve performance and, in some cases, yields slightly lower metric values. This suggests that the more recent architecture does not necessarily translate into better results for this specific task, possibly due to the dataset characteristics or task complexity/meticulousness.

Regarding input representation, the inclusion of answer options leads to marginal improvements in performance, particularly in accuracy and weighted F1-score. This indicates that while the additional context provided by the options is beneficial, its impact is relatively limited, as the question itself already contains most of the relevant information.

In terms of prediction strategy, the *weighted average* approach clearly outperforms the standard *argmax* method in Adjacent Accuracy (± 1), as expected given its design to account for ordinal proximity between classes. However, this comes at the cost of lower performance in strict classification metrics such as accuracy and F1-score, as well as slightly worse results for the H/L order metric. This highlights the trade-off between prioritizing exact level classification and ordinal consistency and reinforces our choice made on Section 3.2.

Macro AUC showed strong values and remained highly consistent across all configurations, with stable class separability regardless of the model, language, or input setup.

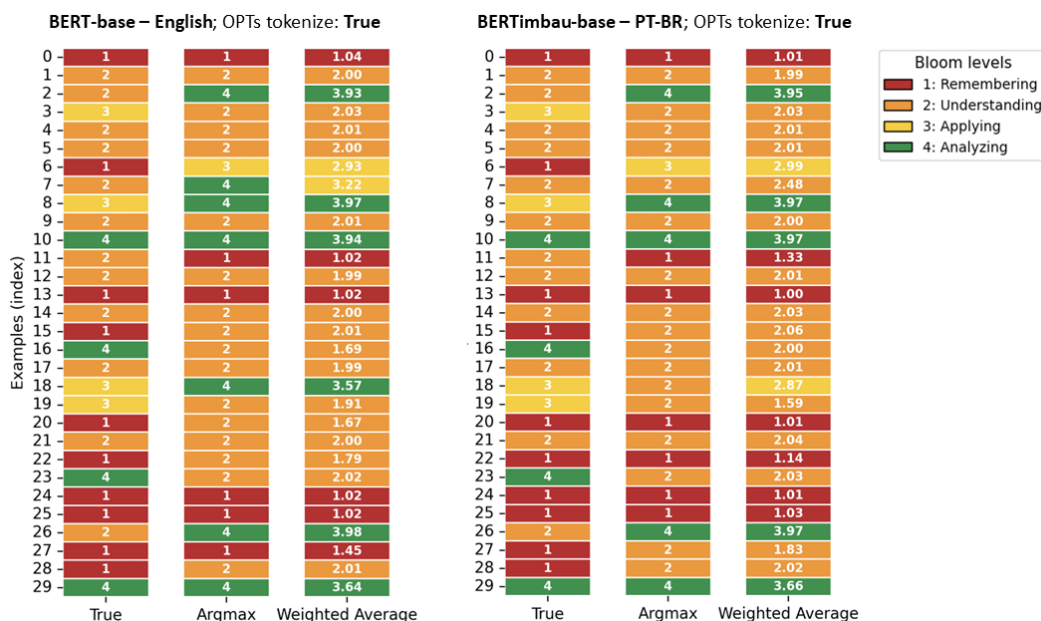
Thus, these results confirm that transformer-based approaches performed way better than the baseline classifier, while still revealing consistent patterns across configurations. We now proceed to a more detailed error analysis, focusing on the BERT-base English and Portuguese variations to better understand the differences in their prediction behavior.

Error analysis between languages

Figure 3 illustrates a side-by-side comparison of predictions for the English and Portuguese BERT-base models (with options tokenized) on the same set of test examples. While overall performance is similar, several instances reveal divergences between the two versions, where one model correctly predicts the Bloom level and the other fails, highlighting the impact of translation on semantic interpretation.

From the 30 samples, instances 7, 18, 20, 22, and 27 exhibit divergences between the English and Portuguese predictions, while instance 8 shows divergence in the prediction strategy for English, and instance 18 for Portuguese. To better illustrate these discrepancies, we revisit the example presented in Figure 1, which corresponds to example 27 in Figure 3. This is a MCQ where the model correctly predicts the original English version as *Remembering*, but classifies the Portuguese version as *Understanding*. This difference is also reflected in the predicted probabilities obtained: the English model assigns 0.56 to *Remembering* and 0.44 to *Understanding*, whereas the Portuguese model assigns 0.17 to *Remembering* and 0.82 to *Understanding*, with the remaining probability mass distributed across the other classes.

This example highlights a subtle cross-linguistic difference that can impact the model prediction. Notably, this example contains none of the main keywords typically used to associate a BT level (Table 2), making the classification more dependent on contextual



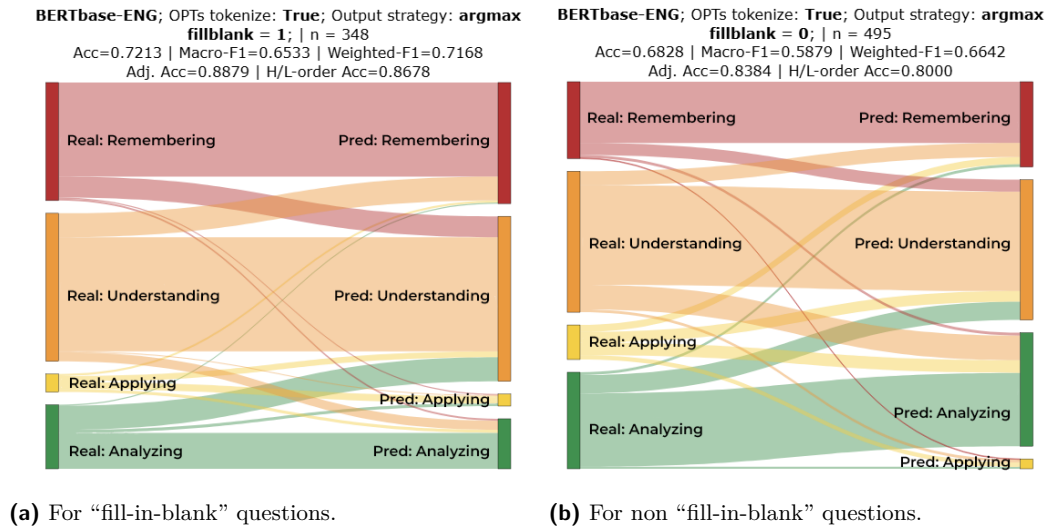
■ **Figure 3** Model output strategies for each language best configuration: 30 examples of test set.

cues than on explicit lexical signals. The Portuguese formulation introduces a syntactic restructuring (“à [FBLANK] da empresa”) that weakens the direct possessive cue present in English (“the firm’s [FBLANK]”), potentially reducing the salience of the underlying economic relationship and making the cognitive level less distinguishable. So, this reformulation may shift the question toward a different classification, as the blank is no longer positioned at the end of the sentence, which may alter the meaning that the model derives from the question. Still, even in the English version, the prediction remains borderline, as reflected in the close probabilities between the two classes, likely due to slight nuances.

Variables Profiling

This analysis is conducted post hoc, by profiling the model predictions according to specific variables, in order to assess whether the model behaves differently across subsets of the data. This was performed for both the fill-in-blank format and the question subject source, indicated by a “Book” field in the original dataset.

For a single run of BERT-base English model with answer options and softmax strategy, performance across fill-in-blank and non-fill-in-blank questions remains largely comparable, with consistent variations in the evaluated metrics: Acc = (0.7213 vs 0.6828) and Weighted F1 = (0.7168 vs 0.6642). From the Sankey plots in Figure 4, most misclassifications occur between consecutive Bloom levels, which is consistent with the relatively high values observed for the Adjacent and H/L Order accuracy metrics. This suggests that predictions tend to remain close to the true class. However, *Applying* remains the most challenging class, often being misclassified with adjacent levels, with notable confusion between *Understanding* and *Analyzing*, likely because instances requiring *Analyzing* also involve *Understanding* (but not vice-versa), as well as similarities in question and answer length and structure. Also, it is important to note that the test set for this profiling variable is unbalanced, with a larger



■ **Figure 4** Sankey Plot and metrics comparison between type of questions (inspired by [12]).

proportion of non-fill-in-blank instances (495 vs 348), which may influence the evaluation, particularly for *Applying*, where the disparity in the number of instances is more pronounced compared to other classes.

Comparing both settings, the fill-in-blank case (Sub-figure 4a) shows slightly more concentrated flows along correct mappings, whereas the non-fill-in-blank case (Sub-figure 4b) exhibits marginally more dispersed flows, particularly between *Understanding*, *Applying*, and *Analyzing*, with notably 61 instances of misclassifications between *Understanding* and *Analyzing* classes. Despite the performance remaining similar across both settings, with only minor differences in metrics and flow distributions, the patterns evidenced may also be influenced by the disparity in the typology of questions.

Across topic-based groupings, the classification performance remains largely consistent, with variations of around 0.02 in the main evaluated metrics. An exception is observed in the “Organizational Behavior” category, where performance deviates more noticeably (a 0.15 variation), likely due to the limited number of samples available for this domain.

5 Conclusion

This study investigated the effectiveness of different approaches for classifying MCQs according to BT, focusing on contextual modeling, input structure, and cross-linguistic robustness. In line with the defined Research Questions (RQs), several key findings emerge.

Regarding RQ1, the results demonstrate that simple rule-based methods are insufficient to capture the complexity of this task, reinforcing the need for models that can benefit from semantic and contextual information. Transformer-based models, specifically BERT and ModernBERT variants, showed significant improvements over the baseline classifier, delivering consistent and robust performance across configurations, with satisfactory metric values. The inclusion of answer options provided marginal gains, suggesting that while additional context is beneficial, the question itself carries most of the discriminative signal. Furthermore, the comparison of prediction strategies highlighted a trade-off between exact classification and ordinal consistency, with the *weighted average* approach improving the created “proximity” metrics at the expense of standard accuracy.

For RQ2, the interlingual analysis revealed that models trained on translated Portuguese data achieve performance comparable to their English counterparts, although with a slight and consistent degradation. A small error analysis demonstrated that even subtle syntactic and structural differences introduced by translation can impact model predictions, particularly in borderline cases where class distinctions are less clear. These findings emphasize the sensitivity of such models to linguistic nuances and still highlight the challenges of maintaining semantic equivalence across languages.

Concerning RQ3, most errors occur between adjacent Bloom levels, particularly *Understanding*, *Applying*, and *Analyzing*, highlighting the difficulty of fine-grained distinctions. Overall, models perform better on *Remembering*, with performance decreasing at the three higher levels mentioned, which is consistent with the literature. This is further influenced by dataset limitations, especially the under-representation of the *Applying* class. Additionally, performance remains stable across formats and the majority of domains, with gains in weighted F1-score for fill-in-blank questions and minor deviations between consecutive levels. Overall, errors are mainly driven by class proximity, data imbalance, and the inherent ambiguity of Bloom-level distinctions.

Beyond these findings, some limitations should be considered. The dataset is heterogeneous and partially uncured, with class imbalance and domain variability, particularly affecting the most underrepresented class. Moreover, the task itself is inherently complex, as BT categories are relatively coarse and often overlapping, making fine-grained distinctions ambiguous. Despite this, the models produce coherent and consistent predictions, with cross-linguistic results showing aligned behavior between English and translated Portuguese data. This indicates that transformer-based approaches capture underlying cognitive patterns that generalize across languages, although performance is still influenced by data quality and the limits of the taxonomy.

Future work may focus on improving both data quality and modeling approaches. On the data side, expanding the dataset with additional MCQs while ensuring a more balanced distribution across Bloom levels, topics, and question types would likely lead to more robust conclusions, as well as support further class-separation analyses, including simplified two-class settings. Additionally, incorporating complementary educational frameworks, such as Item Response Theory (IRT), could provide more nuanced estimates of question difficulty and enable joint modeling of cognitive level and complexity, though this would require student response data [27, 24]. Extending this idea from a modeling perspective, future work could explore hybrid approaches that integrate textual representations with external educational signals, such as IRT-based features or other metadata, to better capture both semantic and psychometric aspects of the task. The integration of external knowledge sources, potentially through retrieval-based mechanisms, may further enhance contextual understanding, while iterative refinement of such hybrid models could improve generalization and robustness. Building on such a pipeline, a promising direction would be the creation of a high-quality Portuguese dataset, for instance, by leveraging LLMs to generate and label MCQs according to BT or richer multi-label schemes, informed by both semantic and difficulty-related signals.

References

- 1 Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

- 2 Fernando R. Altermatt, Andres Neyem, Nicolás I Sumonte, Ignacio Villagrán, Marcelo Mendoza, and Hector J Lacassie. Evaluating the performance of large language models on the conacem anesthesiology certification exam: A comparison with human participants. *Applied Sciences*, 15(11):6245, 2025.
- 3 Lorin W Anderson and David R Krathwohl. *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives: complete edition*. Addison Wesley Longman, Inc., 2001.
- 4 Wen-Chih Chang and Ming-Shun Chung. Automatic applying bloom's taxonomy to classify and analysis the cognition level of english question items. In *2009 Joint Conferences on Pervasive Computing (JCPC)*, pages 727–734, 2009. doi:10.1109/JCPC.2009.5420087.
- 5 Syaamantak Das, Shyamal Kumar Das Mandal, and Anupam Basu. Identification of cognitive learning complexity of assessment questions using multi-class text classification. *Contemporary Educational Technology*, 12(2):ep275, 2020.
- 6 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi:10.18653/v1/N19-1423.
- 7 Amir Hadifar, Semere Kiros Bitew, Johannes Deleu, Chris Develder, and Thomas Demeester. Eduqg: A multi-format multiple-choice dataset for the educational domain. *IEEE Access*, 11:20885–20896, 2023. doi:10.1109/ACCESS.2023.3248790.
- 8 S.S. Haris and Nazlia Omar. Bloom's taxonomy question categorization using rules and n-gram approach. *Journal of Theoretical and Applied Information Technology*, 76:401–407, June 2015.
- 9 Kirk Hillsley. Question format is the best predictor of item discrimination: a multivariable analysis. *Journal of Microbiology and Biology Education*, 26(3):e00205–25, 2025.
- 10 Oliver Holl, Filipe Szolnokyn Cunha, David Streuli, and Timoth   Laborie. Eduquest: Lecture texts and questions for higher education. In Benjamin Paa  yen and Carrie Demmans Epp, editors, *Proceedings of the 17th International Conference on Educational Data Mining*, pages 849–856, Atlanta, Georgia, USA, July 2024. International Educational Data Mining Society. doi:10.5281/zenodo.12729970.
- 11 Fu-Yuan Hsu, Hahn-Ming Lee, Tao-Hsing Chang, and Yao-Ting Sung. Automated estimation of item difficulty for multiple-choice tests: An application of word embedding techniques. *Information Processing & Management*, 54(6):969–984, 2018. doi:10.1016/j.ipm.2018.06.007.
- 12 Kevin Hwang, Kenneth Wang, Maryam Alomair, Fow-Sen Choa, and Lujie Karen Chen. Towards automated multiple choice question generation and evaluation: aligning with bloom's taxonomy. In *International Conference on Artificial Intelligence in Education*, pages 389–396. Springer, 2024. doi:10.1007/978-3-031-64299-9_35.
- 13 Selvia Ferdiana Kusuma, Daniel Siahaan, and Umi Laili Yuhana. Automatic indonesia's questions classification based on bloom's taxonomy using natural language processing a preliminary study. In *2015 International Conference on Information Technology Systems and Innovation (ICITSI)*, pages 1–6. IEEE, 2015.
- 14 Yuheng Li, Mladen Rakovic, Boon Xin Poh, Dragan Gasevic, and Guanliang Chen. Automatic classification of learning objectives based on bloom's taxonomy. In *Proceedings of the 15th International Conference on Educational Data Mining*, pages 530–537. International Educational Data Mining Society, July 2022. doi:10.5281/zenodo.6853191.
- 15 Qian Liu, Nav   Wald, Chandima Daskon, and Tony Harland. Multiple-choice questions (mcqs) for higher-order cognition: Perspectives of university teachers. *Innovations in Education and Teaching International*, 61(4):802–814, 2024. doi:10.1080/14703297.2023.2222715.

- 16 Kseniia Marcq, Erika Jassuly Chalén Donayre, and Johan Braeken. The role of item format in the pisa 2018 mathematics literacy assessment: A cross-country study. *Studies in Educational Evaluation*, 83:101401, 2024. doi:10.1016/j.stueduc.2024.101401.
- 17 Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. mmbert: A modern multilingual encoder with annealed language learning, 2025. doi:10.48550/arXiv.2509.06888.
- 18 Susana Masapanta-Carrión and J. Ángel Velázquez-Iturbide. A systematic review of the use of bloom's taxonomy in computer science education. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education, SIGCSE 2018, Baltimore, MD, USA, February 21-24, 2018*, pages 441–446, February 2018. doi:10.1145/3159450.3159491.
- 19 Alan D. Mead and Chenxuan Zhou. Using machine learning to predict bloom's taxonomy level for certification exam items. *Journal of Applied Testing Technology*, 23:53–71, October 2022. URL: <https://jattjournal.net/index.php/atp/article/view/170341>.
- 20 Nazlia Omar, Syahidah Sufi Haris, Rosilah Hassan, Haslina Arshad, Masura Rahmat, Noor Faridatul Ainun Zainal, and Rozli Zulkifli. Automated analysis of exam questions according to bloom's taxonomy. *Procedia - Social and Behavioral Sciences*, 59:297–303, 2012. Universiti Kebangsaan Malaysia Teaching and Learning Congress 2011, Volume I, December 17 – 20 2011, Pulau Pinang MALAYSIA. doi:10.1016/j.sbspro.2012.09.278.
- 21 Dhaval Patel, Krish Bhikadiya, Nikita Bhatt, Ronakkumar Patel, and Trusha Patel. Impact of bert on evaluating the quality of question papers using bloom's taxonomy. In *International Conference on ICT for Sustainable Development*, pages 65–74. Springer, 2024.
- 22 Radek Pelánek, Tomáš Effenberger, and Jaroslav Čechák. Complexity and difficulty of items in learning systems. *International Journal of Artificial Intelligence in Education*, 32(1):196–232, 2022. doi:10.1007/s40593-021-00252-4.
- 23 Hamdollah Ravand, Reza Shahi, Farshad Effatpanah, and Ali Moghadamzadeh. Measuring cognitive levels in high-stakes testing: A cdm analysis of a university entrance examination using bloom's taxonomy. *Frontiers in Education*, Volume 10 - 2025, 2025. doi:10.3389/feduc.2025.1644811.
- 24 M. Ravikiran, T. Sharma, A. Bhavsar, and R. Saluja. Gisa: Gradual information selection attention for mcq difficulty estimation. *Lecture Notes in Computer Science*, 15877:193–206, 2025. doi:10.1007/978-3-031-98414-3_14.
- 25 Eugénio Ribeiro, Nuno Mamede, and Jorge Baptista. Text Readability Assessment in European Portuguese: A Comparison of Classification and Regression Approaches. In *Proceedings of the International Conference on Computational Processing of Portuguese (PROPOR)*, pages 551–557, 2024. URL: <https://aclanthology.org/2024.propor-1.59/>.
- 26 Ruan Chaves Rodrigues, Jéssica Rodrigues da Silva, Pedro Vitor Quinta de Castro, Nádia Félix Felipe da Silva, and Anderson da Silva Soares. Multilingual transformer ensembles for portuguese natural language tasks. In *Proceedings of the ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese co-located with XII Symposium in Information and Human Language Technology (STIL 2019)*, pages 27–38, March 2020.
- 27 Duden Saepuzaman, Edi Istiyono, Heri Retnawati, et al. Characteristics of two-tier multiple choice (ttmc) high order thinking skill (hots) instruments and the estimation of hots abilities of vocational education students using the item response theory approach. *Journal of Engineering Education Transformations*, pages 13–22, 2025.
- 28 Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. Bertimbau: Pretrained bert models for brazilian portuguese. In Ricardo Cerri and Ronaldo C. Prati, editors, *Intelligent Systems*, pages 403–417, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-61377-8_28.
- 29 J. K. Stringer, Sally A. Santen, Eun Lee, Meagan Rawls, Jean Bailey, Alicia Richards, Robert A. Perera, and Diane Biskobing. Examining bloom's taxonomy in multiple choice questions: Students' approach to questions. *Medical Science Educator*, 31(4):1311–1317, 2021. doi:10.1007/s40670-021-01305-y.

- 30 Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria, July 2025. Association for Computational Linguistics. doi:10.18653/v1/2025.acl-long.127.
- 31 Lamine Zaidi, Karri Grob, Seetha Monrad, Joshua Kurtz, Andrew Tai, Asra Ahmed, Larry Gruppen, and Sally Santen. Pushing critical thinking skills with multiple-choice questions: Does bloom’s taxonomy work? *Academic Medicine*, 93:1, December 2017. doi:10.1097/ACM.0000000000002087.
- 32 James Zhang, Casey Wong, Nasser Giacaman, and Andrew Luxton-Reilly. Automated classification of computing education questions using bloom’s taxonomy. In *Proceedings of the 23rd Australasian computing education conference*, pages 58–65, 2021. doi:10.1145/3441636.3442305.

Inter-Sentential Relations in Enhanced Lexicalized Meaning Representation (E-LMR)

Jorge Baptista 

University of Algarve, Faro, Portugal
INESC-ID Lisboa, Portugal

Abstract

Abstract Meaning Representation (AMR) is a widely adopted framework for graph-based sentence-level semantics. However, its abstraction from surface form and its focus on intra-sentential structure limit its ability to capture discourse-level phenomena. Uniform Meaning Representation (UMR) addresses this limitation by introducing cross-sentential mechanisms, at the cost of increased representational complexity. We argue that inter-sentential relations can be modeled without abandoning lexical anchoring. Within Lexicalized Meaning Representation, we propose a programmatic extension grounded in three domains: (i) nominal and pronominal anaphora, (ii) temporal anchoring, and (iii) inter-sentential cohesion devices. We call this Enhanced LMR (E-LMR). We contend that these relations are not abstract add-ons, but are lexically realized and should be represented accordingly. Rather than a full annotation scheme, we introduce design principles and representation strategies illustrated with European Portuguese data. A lexically grounded approach improves interpretability, annotation consistency, and cross-linguistic robustness. We thus position E-LMR as a viable alternative within a modular semantic architecture where discourse relations are explicitly encoded while remaining tightly coupled to lexical form.

2012 ACM Subject Classification Computing methodologies → Discourse, dialogue and pragmatics

Keywords and phrases Inter-sentential relations, Lexicalized Meaning Representation, Nominal and Pronominal Anaphora Resolution, Temporal Anaphora Resolution, Inter-Sentential Cohesion Devices, European Portuguese

Digital Object Identifier 10.4230/OASICS.SLATE.2026.5

Supplementary Material *Dataset:* <https://doi.org/10.13140/RG.2.2.29402.50888>

Funding *Jorge Baptista:* Work supported by national funds through Fundação para a Ciência e a Tecnologia, I.P. (FCT) under projects UID/50021/2025 (DOI: <https://doi.org/10.54499/UID/50021/2025>) and UID/PRR/50021/2025 (DOI: <https://doi.org/10.54499/UID/PRR/50021/2025>).

1 Introduction

Abstract Meaning Representation (AMR) has been central to the development of graph-based semantic representations, providing a structured and largely language-agnostic encoding of sentence-level meaning [1, 2, 22, 27]. In this framework, meaning is primarily located in the domain of predication, that is, in the relations between a predicative operator and its arguments. Core semantic relations are encoded by `:argi`-labeled edges, while non-core relations capture circumstantial information such as location, time, cause, or concession. However, its design principles – particularly its focus on intra-sentential structure and its abstraction from surface lexical material – limit its ability to capture discourse-level phenomena. Recent extensions, most notably Uniform Meaning Representation (UMR) [11, 12, 13, 14, 15, 28], address these limitations by introducing mechanisms for cross-sentential linking, including coreference, temporal anchoring, and discourse relations.

Within this evolving landscape, we argue for a complementary, lexically grounded approach to inter-sentential semantics within Lexicalized Meaning Representation (LMR) [7, 9, 10]. While LMR prioritizes lexical realization and sentence-level anchoring, it lacks a fully



© Jorge Baptista;
licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 5; pp. 5:1–5:18

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

articulated model for representing relations across sentence boundaries. We therefore propose a programmatic extension, Enhanced LMR (E-LMR), designed to capture inter-sentential relations while preserving lexical transparency and structural interpretability.

We focus on three domains: (i) nominal and pronominal anaphora, (ii) temporal anchoring and temporal anaphora [17, 21], and (iii) inter-sentential cohesion devices, including discourse adverbials and connective structures. Building on insights from UMR [13, 28], we provide preliminary evidence how these phenomena can be represented in a lexically anchored framework using PENMAN graphs [16], without excessive abstraction or loss of alignment with surface form.

In this framework, both in AMR and LMR, sentences are represented as PENMAN graphs, lexical units are represented as graph nodes, while syntactic-semantic relations are encoded as labeled edges. Predicative elements introduce argument structures, typically represented through `:argi` relations, while modifiers, possession, coordination, and co-reference are represented through dedicated edge labels. The representation remains closely aligned with surface lexical realization (see Fig. 1 for an illustration). Particular attention is given to lexical items, such as pronouns, temporal expressions, and discourse markers, as carriers of inter-sentential relations. The E-LMR extension to capture discourse relations also uses the same PENMAN graph formalism, making integration of sentence-level annotations straightforward.

Although the examples discussed here are drawn from European Portuguese, the proposed mechanisms are not language-specific. On the contrary, the explicit lexical anchoring adopted in E-LMR is expected to facilitate adaptation to morphologically rich languages, where reference and discourse relations are frequently encoded through inflectional or function-word material.

Rather than proposing a fully formalized annotation scheme, this paper advances a principled set of design hypotheses and representation strategies, illustrated with examples from Portuguese. We argue that a lexically grounded approach to discourse representation offers advantages in interpretability, annotation consistency, and cross-linguistic adaptability, especially for languages with rich morphological and syntactic variation, such as Portuguese.

We position E-LMR as a lexically grounded alternative within a modular semantic architecture, where sentence-level representations remain text-anchored and inter-sentential relations are explicitly encoded. This contributes to ongoing discussions on the balance between abstraction and lexical specificity and on the role of discourse-level semantics in NLP.

The remainder of the paper is structured as follows. Section 2 motivates and defines E-LMR, focusing on nominal and pronominal anaphora, temporal anchoring, and discourse relations. Section 3 presents the annotation methodology. Section 4 reports the main results. Section 5 concludes and outlines future work.

2 Discourse modeling: Multiple relations across sentences

A variety of linguistic devices contribute to the cohesion and coherence that turn a sequence of sentences into a discourse, that is, a meaningful and interpretable text [18, 19, 20]. These include utterance structure, dialogue organization, pragmatic functions, speakers' intentions, and cohesive mechanisms such as reference, ellipsis, conjunction, and lexical relations.

These dimensions operate at different levels of abstraction and with varying degrees of explicit linguistic realization. In this paper, we focus on inter-sentential cohesion devices, which provide a linguistically grounded entry point for modeling discourse relations within semantic representation frameworks.

To this end, we argue for a programmatic extension of the Lexicalized Meaning Representation (LMR) framework [9], aimed at systematically capturing inter-sentential relations within a lexically grounded approach to discourse semantics. We call this *Enhanced-LMR* (E-LMR).

Originally inspired in Abstract Meaning Representation (AMR) [2] for encoding intra-sentential relations, LMR remains limited if confined to the sentence level. We contend that any adequate meaning representation must extend to inter-sentential structure, where reference, temporality, and discourse connectivity are constructed and resolved.

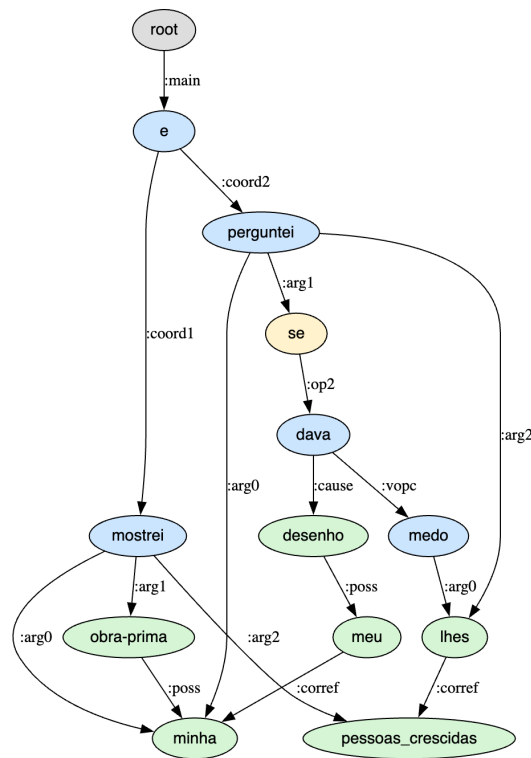
In this respect, the extension from intra- to inter-sentential representation is not merely desirable, but theoretically necessary. Still, while E-LMR relies on lexical anchoring, the relations encoded are not purely lexical associations. Rather, lexical items function as explicit access points to discourse-level semantic interpretation, including reference resolution, temporal anchoring, and discourse connectivity. Certain inferential relations requiring extensive world knowledge or pragmatic reasoning remain, however, outside the scope of the current framework.

This position aligns with recent developments extending AMR into Uniform Meaning Representation (UMR), which incorporate cross-sentential phenomena such as coreference, temporal anchoring, and discourse relations [14, 11, 12, 13, 15, 28]. However, while UMR advances discourse modeling through increasingly abstract and layered representations, it tends to weaken the link to surface lexical realization. In contrast, LMR maintains a principled commitment to lexical transparency and text anchoring. It therefore requires an alternative path to inter-sentential modeling – one that encodes discourse relations explicitly while preserving their lexical grounding. This motivates the approach proposed in this paper.

2.1 Inter-sentential, nominal and pronominal, anaphora resolution

The LMR guidelines [9] adopt a minimal annotation strategy aimed at reducing annotator cognitive load by encoding only elements explicitly realized in the sentence, almost without reconstructing zeroed material. Still, controlled reconstruction of linguistically recoverable zeroed material is proposed in [3]. Part-of-speech tagging and word-sense disambiguation are deferred to a post-processing stage unless required for specific annotation decisions. For word-sense disambiguation of predicative elements, we rely on the Lexicon-Grammar of European Portuguese [4, 5, 6, 7].

In both AMR and LMR, intra-sentential anaphora is resolved at the annotation level by assigning multiple semantic relations to a single variable, instantiated only once in the sentence graph. This is illustrated in Fig.1. (Unless otherwise indicated, examples are drawn from the European Portuguese translation of *O Príncipezinho*, based on the French original of Antoine de Saint-Exupéry’s *Le Petit Prince* (*The Little Prince*, 1943). Examples are identified by sentence id; English translations are our own.) In this Figure, several edges point to the node *pessoas crescidas* “grown-ups”. As in many null-subject languages, Portuguese does not systematically realize grammatical subjects at the surface level. This is the case for the first-person singular subjects of *mostrei* “showed” and *perguntei* “asked” (here referring to the speaker/narrator). In this sentence, however, the referent is overtly realized by the possessive pronoun *minha* “my_fem.sg.”, and the zeroed subjects are therefore linked (*arg0*) to this explicit node. The possessive *meu* “my_masc.sg.” co-refers to the same entity, yielding an explicit *:corref* relation. In both cases, these pronouns refer to the intra-diegetic narrator. Similarly, the pronoun *lhes* “to_them” is linked to *pessoas crescidas* “grown-ups” via a *:corref* relation. While narrator reference operates at the discourse level, the resolution of *lhes* remains local to the sentence.



■ **Figure 1** LMR graph of the sentence: *Mostrei a minha obra-prima às pessoas crescidas e perguntei-lhes se o meu desenho lhes dava medo* [opz-id-10b] “I showed my masterpiece to the grown-ups and asked them if my drawing scared them.”

Inter-sentential reference resolution is significantly more complex. In what follows, we outline the main components of this task.

We propose that each sentence-level LMR graph remains an autonomous representation, while being embeddable within a higher-level discourse graph expressed in PENMAN format. At the discourse level, a root node DISCOURSE provides a global anchor and introduces a small set of (default) metalinguistic parameters, to encode speaker (SPKR), addressee (ADRS), location (HERE), and utterance time (NOW), and serve as stable referential anchors across sentences.

Sentence-level graphs are inserted into the discourse graph in textual order, determined by their identifiers. At an initial stage, they are linked by a generic relation such as `:and`, encoding linear succession while leaving the precise discourse relation underspecified.

This structure enables a second processing layer in which the discourse graph is enriched with inter-sentential links. For example, a zeroed first-person subject can be resolved via a `:corref` relation linking it to SPKR. Similarly, pronominal, nominal, spatial, and temporal references can be connected to discourse anchors or antecedents introduced in previous sentences. Sentence-level annotation thus remains simple, while discourse-level interpretation is handled incrementally through graph enrichment.

The following pseudo-code illustrates this architecture.

```
(d / discourse
  :SPKR (SPKR / speaker)
  :ADRS (ADRS / addressee)
  :HERE (HERE / here)
  :NOW (NOW / now)
  :AND (s1 / sentence
        :id "10a"
        :content ( ...graph for sentence 10a... ))
  :AND (s2 / sentence
        :id "10b"
        :content ( ...graph for sentence 10b... ))
  :AND (s3 / sentence
        :id "10c"
        :content ( ...graph for sentence 10c... )))
```

Once sentence id="10b" is integrated into the discourse graph, the first-person reference can be resolved by introducing a `:corref` relation linking it to SPKR, e.g., `(m / minhas :corref SPKR)`. This architecture provides a lexically grounded alternative to discourse-level representation, avoiding the need for abstract intermediate layers.

In the case of a simple zeroed subject, as illustrated in sentence (1):

(1) *Voei um pouco por todo o mundo.* [id=opz-22-1] “I flew all over the world.”

the initial LMR annotation therefore contains only the overt material:

```
(root :main (v / voei
  :path (p / por :op2 (m / mundo
    :quant (t / todo
      :quant (up / um_pouco))))))
```

After post-processing, the graph is enriched with lemmatized, morphosyntactic, and word-sense information, yielding the E-LMR representation shown below:

```

(root
  :main (v / voei
    :lemma (lem1 / voar)
    :pos (pos1 / verb
      :tense (ten1 / past)
      :mode (mod1 / ind)
      :pers (per1 / 1)
      :nb (num1 / sg))
    :wsd (wsd1 / 37LD)
    :arg0 SPKR
    :path (prep1 / por
      :op2 (m1 / mundo
        :lemma (lem2 / mundo)
        :pos (pos2 / noun
          :subcat (sub1 / common)
          :gen (gen1 / m)
          :nb (num2 / sg))
        :quant (t1 / todo
          :lemma (lem3 / todo)
          :pos (pos3 / pron
            :subcat (sub2 / indef)
            :gen (gen2 / m)
            :nb (num3 / sg))))))
    :quant (up1 / um_pouco
      :lemma (lem4 / um_pouco)
      :pos (pos4 / adv
        :subcat (sub3 / mq-i))))))

```

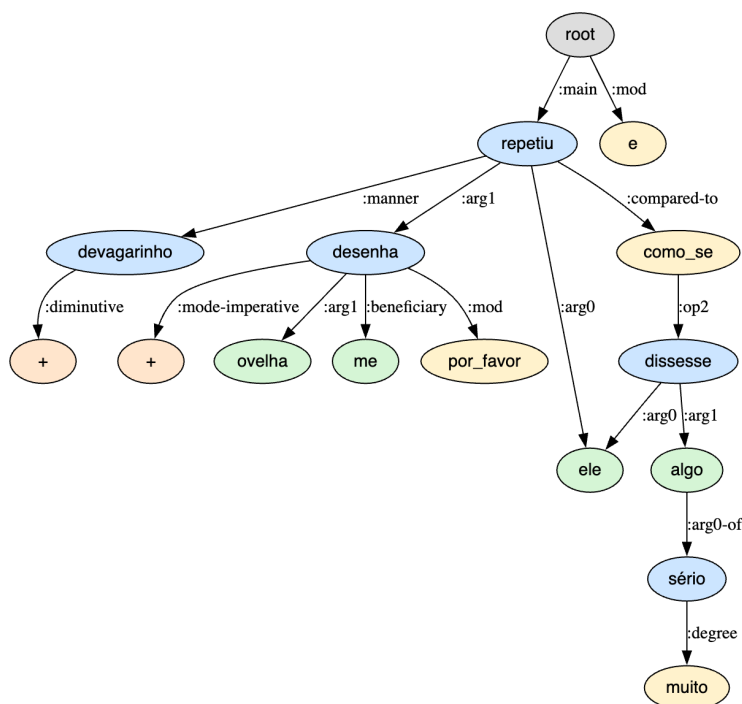
In this representation, the zeroed subject of *voar* “fly” is explicitly instantiated and resolved as coreferential with the discourse speaker (SPKR). Under the imperative mode, as in (2), the zeroed addressee (*você*s “You_pl.”) can be reconstructed by linking the verb to ADRES, e.g., (*i* / *imaginem* :arg0 ADRES). Similarly, the first-person singular (*eu* “I”) zeroed passive agent is linked to the narrator, e.g., (*d* / *despertado* :arg1 SPKR).

- (2) *Assim, imaginem [você]s a minha surpresa quando, ao amanhecer, [eu] fui despertado por uma vozinha muito curiosa.* [id=opz-43] “So, imagine my surprise when, at daybreak, I was woken up by a very curious little voice.”

We further exploit direct-speech annotation to distinguish narrator-level utterances from character-level utterances, which may occur in either reported (indirect) or direct speech. Consider the next example (3) and the corresponding LMR graph, presented in Fig.2.

- (3) *E ele repetiu, devagarinho, como se dissesse algo muito sério: – Por favor... desenha-me uma ovelha...* [id=opz-60] “And he repeated, softly, as if he were saying something very serious: – Please... draw me a sheep...”

Since the embedded clause is in direct speech (here in the imperative) the beneficiary is interpreted as coreferential with the subject of the “verb of saying” (*verbum dicendi*) *repetir* “repeat”. This licenses the insertion of the edge (*m* / *me* :corref *e2*), where *e2* denotes *ele* “he”. As for the subject of *desenhar* “draw”, a subsequent annotation module reconstructs the implicit argument slot, yielding (*d* / *desenha* :arg0 SPKR).



■ **Figure 2** LMR graph of the sentence OPZ-ID-60: *E ele repetiu, devagarinho, como se dissesse algo muito sério: – Por favor... desenha-me uma ovelha... “And he repeated, softly, as if he were saying something very serious: – Please... draw me a sheep...”*

The more general case of inter-sentential *pronominal anaphora* resolution is illustrated in (4)–(5). In such cases, the antecedent of the pronoun *ele* in (5) typically occurs in a nearby sentence, in this case, *rapazinho* in (4). Unresolved anaphors are those lacking an explicit **:corref** relation. An automatic anaphora resolution module [8, 23, 26] may be applied at this stage, with subsequent manual correction.

- (4) *E vi um **rapazinho** extraordinário a fitar-me com ar sério.* [id=opz-50] “And I saw an extraordinary little boy looking at me seriously.”
- (5) *Este é o melhor retrato que, mais tarde, consegui fazer dele.* [id=opz-51] “This is the best portrait [that] I was able to make of him later.”

Lexical anaphora, where a noun refers to a previously introduced entity, poses a greater challenge. In (6), the noun *aparição* “apparition” refers to *rapazinho*, also in (4).

- (6) *Olhei para aquela **aparição** com os olhos arregalados de espanto.* [id=opz-55] “I looked at that apparition with wide eyes of astonishment.”

Such cases typically require world knowledge and contextual interpretation, and are therefore less amenable to automatic resolution, demanding human intervention. This case is then annotated as (a / **aparição :corref id-50_r**), where the compound reference **id-50_r** stands for the previous occurrence of *rapazinho*, and consists of the sentence id and the variable instantiating the antecedent noun.

Finally, certain omitted arguments are reconstructed from lexical information available in the Lexicon-Grammar entries of the relevant predicates. This occurs in the two consecutive sentences shown in Examples (7)–(8), where the unexpressed subject of *representar* “represent”

in (8) refers to the same discourse entity as the subject of the previous occurrence of that verb in (7). To represent this dependency explicitly, an artificial [zero] node is introduced in the corresponding argument position and linked to its antecedent through a coreference relation.

- (7) *O meu **desenho** não representava um chapéu.* [id=opz-13] “My drawing did not depict a hat.”
- (8) *Representava uma jiboia a digerir um elefante.* [id=opz-14] “It depicted a boa constrictor digesting an elephant.”

```
(root :main (r / representava
  :arg0 (z / [zero] :corref id-13_d)
  :arg1 (j / jiboia
    :arg0-of (d / digerir
      :arg0 j
      :arg1 (e / elefante))))))
```

At this stage, we envisage an annotation interface that performs: (i) automatic lemmatization, PoS tagging, and word-sense disambiguation (with human-in-the-loop when needed); (ii) insertion of placeholder nodes and edges based on Lexicon-Grammar argument structure; (iii) resolution of references to discourse anchors (SPKR, ADRS, HERE, NOW); and (iv) manual resolution of remaining anaphora.

The proposed framework is intended as a semi-automatic enrichment architecture. Several components are amenable to automatic processing, including lemmatization, morphosyntactic tagging, argument-slot instantiation based on Lexicon-Grammar resources, temporal expression recognition, and candidate antecedent detection for anaphora resolution. The resulting discourse graph can therefore be incrementally enriched through a pipeline combining rule-based modules, statistical resolvers, and human validation.

2.2 Inter-sentential temporal anchoring and temporal anaphora

Temporal anaphora involves identifying events, analyzing tense and mode, handling clause structure, and interpreting temporal modification [21, 29, 30, 31, 32].

In LMR, temporal expressions are annotated as single nodes with basic semantic types: *date* (temporal location), *duration*, and *frequency* (combining modulo and duration values). Examples, taken from the novella, include *ao amanhecer* “at dawn”, *mais tarde* “later”, *em 1920* “in 1920” (date); *para sempre* “forever” (duration); and *às vezes* “sometimes” (frequency). For temporal anaphora, *date* expressions are the most relevant. A temporal named-entity recognition (NER) system should be able to identify and annotate such expressions with their corresponding types and normalized values [17].

Consider the temporal expression *ao amanhecer* “at dawn”, already appearing in (2). It is annotated as `NE_TEMPO_T-DATE_T-REF-ENUNC(a_o amanhecer)`, indicating a referential-relative temporal expression anchored to the time of enunciation. The event *despertar* “wake” is thus located at a time of day whose precise reference depends on its relation to the time of enunciation. Since the temporal expression and the event occur in the same sentence, the annotation reduces to encoding its semantic type and reference.

```
... :time (qd / quando :op2 (d / despertado
  :vaux (f / fui)
  :time (aa / ao_amanhecer
    :lemma (lem1 / ao_amanhecer)
```



```

      :pos (pos1 / adv
        :type (type / date)
        :ref (ref / enunc)))
    ...

```

When the antecedent of a temporal expression occurs in a previous utterance, the annotation links it to the corresponding discourse segment. For example, in (9), the expression *nesse momento* “at that moment” refers to a preceding stretch of dialogue and is encoded via a *t-ref* relation to that segment.

- (9) *Nesse momento, um clarão iluminou o mistério da presença dele, o que me fez perguntar bruscamente: – Então vieste de outro planeta?* [id=opz-123] “At that moment, a flash of insight illuminated the mystery of his presence, which made me abruptly ask: – So, you came from another planet?”

```

(root :main (i / iluminou
  :time (nm / nesse_momento
    :lemma (lem1 / nesse_momento)
    :pos (pos1 / adv
      :type (type / date)
      :ref (ref / enunc :t-ref id-121-id-122)))
  ...

```

Consider now (10) and the temporal expression *mais tarde* “later”, modifying the event *fazer um retrato* “make a portrait”.

- (10) *Este é o melhor retrato que, **mais tarde**, consegui fazer dele.* [id=opz-51] “This is the best portrait [that] I was able to make of him later.”

The event expressed by the support-verb construction *fazer retrato* “make portrait” in (4) is located *after* some narrated events and *before* the completion of the narration as a whole, values that are encoded by *t-after* and *t-before*.

```

(root :main (r1 / retrato
  ...
  :vsup-of (f / fazer
    :vaux (c / consegui)
    :time (mt / mais_tarde
      :lemma (lem1 / mais_tarde)
      :pos (pos1 / adv
        :type (type / date)
        :ref (ref / enunc :t-after id-50 :t-before NOW)))
  ...

```

While *t-before* is straightforward, determining the scope of *t-after* is more difficult. By default, it is linked to the immediately preceding sentence.

Absolute dates, such as *em 1920* “in 1920”, are mapped to normalized temporal values (simplified here as [value]).

- (11) *O astrónomo repetiu a sua apresentação **em 1920**, vestido com um fato muito elegante.* [id=opz-158] “The astronomer repeated his presentation in 1920, dressed in a very elegant suit.”

```
(root :main (a1 / apresentação
  :vsup (r / repetiu)
  :arg0 (a / astrónomo)
  ...
  :time (d / em_1920
    :type (type / date)
    :ref (ref / absol :t-ref [value])))
```

These mechanisms provide evidence that temporal anchoring can be modeled within a lexically grounded framework through explicit linking, without requiring abstract temporal layers.

2.3 Other inter-sentential cohesion devices

Conjunctive adverbs are a major device of inter-sentential cohesion [24, 25], e.g., *assim* “thus”, *consequentemente* “consequently”, *contudo* “nevertheless”, *então* “then”, *no entanto* “however”, and *portanto* “therefore”. In LMR, such sentence-external modifiers are represented as modifiers of the **root** node, enabling inter-sentential representation. Each adverb is associated with a semantic tag drawn from 13 categories, including *additive*, *concessive*, *conclusive*, *consecutive*, *continuative*, *contrastive*, *exemplificative*, and *temporal*.

Consider the conjunctive adverbs *mesmo assim* “even so/yet”, *pois* “indeed”, and *no entanto* “however”, illustrated in the following dialogue, examples (12)–(14):

- (12) – *As pessoas do teu planeta – disse o príncipezinho – cultivam cinco mil rosas num único jardim... e **mesmo assim** não encontram aquilo que procuram...* [id=opz-1307] ‘ – The people on your planet – said the little prince – grow five thousand roses in a single garden... and yet they do not find what they are looking for...’
- (13) – ***Pois**, não encontram... – repliquei.* [id=opz-1308;1309] ‘ – Indeed, they do not find it... – I replied.’
- (14) – ***No entanto**, o que procuram pode ser encontrado numa só rosa ou num pouco de água...* [id=opz-1310] ‘ – And yet, what they are looking for can be found in a single rose or in a little water...’

These adverbs are represented in LMR as modifiers of the **root** node or of the highest predicate in the clause they modify. Crucially, however, this representation does not encode any explicit link between the current sentence and preceding discourse segments.

In E-LMR, by contrast, discourse relations are explicitly encoded through the combination of the adverb’s syntactic-semantic class (**advconj**), its semantic category, and dedicated discourse-linking relations. Specifically, a **d-ref** edge connects the current sentence (or predicate) to the relevant element in the preceding discourse, while **d-ref-type** specifies the nature of the semantic relation.

```
(root :main (d / disse ...
  :arg1 (e1 / e :direct-speech +
    :coord1 (c / cultivam ...
    :coord2 (e2 / encontram
      :mod (ma / mesmo_assim
        :lemma (lem1 / mesmo_assim)
        :pos (pos1 / adv
          :subcat (sub1 / advconj
            :d-ref-type (type1 / concessive)
            :d-ref c )))
    ...
```

In this example, representing (12), the adverb *mesmo assim* “even so/yet” links the main predicate of the second coordinated clause (*encontram* “find”) to that of the first (*cultivam* “grow”), establishing a *concessive* relation. The adverb thus operates within a single syntactic sentence.

Considering example (13) and the adverb *pois* “indeed”:

```
(root :main (r / repliquei ...
  :arg1 (e / encontram :direct-speech +
    :mod (p / pois
      :lemma (lem1 / pois)
      :pos (pos1 / adv
        :subcat (sub1 / advconj
          :d-ref-type (type1 / justifier)
          :d-ref id-1307-e2)))
    ...
```

In this example, the adverb *pois* links *encontram* “find” to its previous occurrence in the preceding sentence, identified by the sentence id-1307 and the variable e2 within.

Finally, regarding example (14) and the adverb *no entanto* “yet/however”:

```
(root :mod (ne / no_entanto
  :lemma (lem1 / no_entanto)
  :pos (pos1 / adv
    :subcat (sub1 / advconj
      :d-ref-type (type1 / contrastive)
      :d-ref id-1307)))
  ...
```

The adverb *no entanto* “yet/however”, represented as a modifier of the **root** node, links the entire sentence to a preceding utterance of the same speaker, establishing a *contrastive* relation.

In these examples, adverbs establish relations between distinct utterances. While *pois* links consecutive sentences, *no entanto* connects the current sentence to a non-adjacent one.

Another major cohesive device is the use of conjunctions such as *e* “and” (15) and *mas* “but” (16) at the beginning of sentences, where they function similarly to conjunctive adverbs. We therefore reclassify these elements as conjunctive adverbs and assign them discourse functions, namely *continuative* and *contrastive*.

(15) *E a geografia, é verdade, ajudou-me muito.* [id=opz-22-2] “And geography, it’s true, helped me a lot.”

(16) *Mas, tal como os anteriores, foi rejeitado.* [id=opz-85] “But, like the previous ones, it was rejected.”

In this case, we reclassify these conjunctions as adverbs and assign them the corresponding discourse functions, namely *continuative* and *contrastive*, respectively.

These cases indicate that discourse relations can be directly encoded through lexical items and their distribution, supporting a lexically grounded approach to inter-sentential semantics.

3 Evaluation

To evaluate the proposed E-LMR extension, we consider four dimensions: interpretability, lexical grounding, discourse coverage, and annotation feasibility. These dimensions were assessed qualitatively through manual inspection of the enriched graphs and quantitatively

through counts of introduced relations, reconstructed arguments, and discourse-level links. Rather than benchmarking against existing frameworks, this preliminary study aims to determine whether the proposed mechanisms can be systematically implemented within a lexically grounded representation. Given the absence of comparable UMR annotations and mature annotation tools, we conducted an exploratory annotation experiment focused on assessing the feasibility of lexically grounded discourse enrichment.

To this end, we selected the first two chapters of the European Portuguese version of *O Príncipezinho*. This small corpus comprises 102 sentences, including both narrative passages and dialogue. We leveraged Lexicon-Grammar resources for verbs and predicative nouns to perform word-sense disambiguation and to instantiate zeroed argument slots (e.g., subjects and essential complements).

The annotation process followed a staged enrichment procedure. Annotators first produced sentence-level LMR graphs based on overt lexical material.¹ In a second stage, these graphs were enriched with lexical, morphosyntactic, and discourse-level information, including reconstructed argument slots, co-reference relations, temporal anchoring, and discourse links.²

We then enriched the LMR graphs to comply with the E-LMR guidelines described in Section 2, introducing new nodes and edges for inter-sentential features, including co-reference :**corref**, temporal reference :**t-ref**, and discourse relations **d-ref**. Grammatical features were not the focus, as they can be added mechanically.

This setup allows us to assess how E-LMR captures inter-sentential structure while preserving lexical anchoring, and to evaluate its feasibility on a linguistically rich, albeit small, corpus.

The annotation was performed independently by two linguists familiar with LMR, using task-specific guidelines. Given the exploratory nature of the study and the ongoing refinement of the guidelines, disagreements were resolved through adjudication and inter-annotator agreement is left for future work.

Differences arose mainly from: (i) omitted argument slots; (ii) missing :**corref** relations; (iii) uncertainty in connective’s semantics; (iv) ambiguity in anaphoric or discourse targets; and (v) formal inconsistencies (e.g., indentation, spacing, typos).

Given the exploratory nature of this study and the consensus achieved between annotators, inter-annotator agreement is not reported at this stage. This will be addressed in future work once the guidelines and annotation tools have reached a more stable level of maturity.

4 Results

We now present the main results of the annotation task, followed by a discussion of selected problematic cases.

Arguments. The LMR corpus contains 306 argument relations, including 38 :**arg_i-of** relations (relative subclauses and adjectival modifiers). We introduced 66 additional relations, yielding a total of 372 argument slots. Of the instantiated argument slots, 55 are assigned to SPKR, 8 to previously introduced variables, and 3 to ADRS. These figures suggest that a substantial portion of the semantic connectivity in the text emerges from referential links extending across sentence boundaries, forming a dense discourse structure that is not fully

¹ <https://doi.org/10.13140/RG.2.2.15980.73607>

² <https://doi.org/10.13140/RG.2.2.29402.50888>

captured by sentence-level representations alone. These results also provide preliminary evidence that inter-sentential relations can be systematically encoded within a lexically grounded framework.

Co-reference. Regarding coreference, 31 nodes already carried `:corref` relations, while 89 were newly introduced (32 to SPKR, 57 to previous nodes). The predominance of SPKR-oriented links reflects the narrative structure of the text, in particular its first-person perspective and is likely to differ across genres. Argument omission at the surface level is, then, relatively frequent in this corpus. When a coreferential anaphor is overtly realized, omitted arguments are already represented in LMR. Otherwise, artificial **zero** nodes are introduced (22 cases).

Temporal reference. Temporal reference is annotated only when triggered by explicit temporal expressions, primarily temporal adverbs and temporal named entities. Temporal interpretation derived from tense and mood is not addressed at this stage and is left for future work. The corpus contains 14 `:time` relations: 5 involving adverbs and 9 involving phrases or subclauses introduced by temporal prepositions or conjunctions. No absolute referential dates were found. All relative temporal expressions are anchored to the enunciation, typically to the main event of the clause.

Discourse relations. Discourse relations are annotated primarily when signalled by explicit connective elements. A total of 19 `:d-ref-type` relations were introduced: 1 conclusive, 5 consecutive, 8 continuative, and 5 contrastive. In addition, 6 instances of `:direct-speech` + mark clauses realized in direct speech, typically under a *verbum dicendi*, and 2 instances of `:direct-speech-cont` + indicate continuation within the same discourse turn.

These results provide preliminary evidence that discourse relations can be systematically encoded through overt lexical cues, reinforcing a lexically grounded approach to inter-sentential semantics.

The annotation process also revealed systematic limits of text-internal discourse representation, particularly in cases involving underspecified deixis, extra-textual reference, and world-knowledge-dependent interpretation.

Indefinites. One noteworthy case concerns argument positions with *indefinite* reference (4 instances), as exemplified in (17). Although verbal inflection licenses the reconstruction of a first-person plural subject *nós* “we”, the recovered argument does not refer to a specific plural entity or group. Rather, it receives a generic interpretation, denoting an indefinite set of individuals that includes the speaker as a representative member.

- (17) *Elas precisam sempre que [nós] lhes explicuemos as coisas.* [id=opz-16] “They always need us to explain things to them.”

This is the same generic use of first-person plural *nós* “we” that appears overtly in sentences such as (18).

- (18) *Quando um mistério nos deixa demasiado espantados, não nos atrevemos a desobedecer.* [id=opz-61] “When a mystery leaves us too astonished, we do not dare to disobey.”

In other cases, the indefinite referent corresponds to a zeroed argument (19), recovered from the predicate syntactic-semantic represented in the Lexicon-Grammar.

- (19) *Responderam-me: “Porque é que um chapéu haveria de dar medo [a alguém]?”*
[id=opz-11-12] “They replied to me, ‘Why would a hat be scary?’”

This lead us to introduce an INDEF node in the annotation scheme: (z / [zero] :corref INDEF), for the zeroed arguments (17) and (19), and (n / nos :corref INDEF) for the overt pronouns of (18).

These cases constitute evidence that indefiniteness can be encoded through morphosyntactic cues without requiring explicit discourse anchors.

Not-defined. In a number of cases (12), no clear antecedent could be assigned to an explicit anaphor, highlighting limits of strictly text-internal reference resolution.

- (20) *Que era assim:* [id=opz-10a] “It went like this:”
(21) *Aqui fica uma cópia dessa gravura.* [id=opz-4] “Here is a copy of that illustration.”

In these cases, deictic expressions refer to extra-textual objects/entities (e.g., illustrations) rather than to discourse-internal nodes. These cases are represented by a **not-defined** node.

A more complex situation is illustrated in (22):

- (22) *Quando finalmente consegui falar, perguntei-lhe: – Mas... que estás a fazer aqui?*
[id=opz-59] “When I finally managed to speak, I asked him: – But... what are you doing here?”

Although the previously introduced noun *deserto* “desert” could in principle serve as an antecedent for the adverb *aqui* “here”, its reference remains underspecified and does not clearly map to the discourse-level variable *HERE*.

In other cases, the pronoun *isso* in expressions such as *por isso* “because of that/thus” cannot be straightforwardly mapped onto a specific discourse node, segment or discourse span, reflecting a form of underspecified or diffuse reference, as in example (23).

- (23) *Por isso, tive de escolher outra profissão, e aprendi a pilotar aviões.* [id=opz-21]
“So, I had to choose another profession and learned to fly airplanes.”

In this context, the pronoun refers to the reaction of the *peessoas crescidas* “grown-ups” to the narrator’s attempts to pursue a drawing career, which had been elaborated across an extended stretch of the preceding discourse.

Similarly, some reconstructed **zero** nodes lack a clear referential target, like in example (24):

- (24) *Ele analisou com toda a atenção e disse: – Não!* [id=opz-76] “He examined it very carefully and said: ‘No!’”

In this instance, the object of *analisar* “analyse” can be interpreted as referring to a drawing, even though no suitable antecedent is found in the immediately preceding discourse. Instead, this interpretation is inferred from the overall dialogue context, where the Little Prince repeatedly requests drawings and the narrator successively produces a series of figures. These cases are therefore encoded as **not-defined**, deferring resolution to a later stage.

These phenomena highlight the limits of strictly text-anchored representations, as some discourse relations can only be resolved by drawing on contextual or world knowledge that is not explicitly encoded in the text.

The results of the annotation experiment provide preliminary support for the four evaluation dimensions introduced above. In terms of *interpretability*, the use of lexically grounded nodes and explicit relations (e.g., `:corref`, `:d-ref`, `:t-ref`) yields representations that remain closely aligned with the surface text, facilitating human inspection and manual validation.

Regarding *lexical grounding*, the analysis confirms that a substantial portion of inter-sentential relations – particularly anaphora and discourse connectivity – can be directly anchored in overt lexical material, such as pronouns, temporal expressions, and conjunctive adverbs.

With respect to *discourse coverage*, the enrichment process suggests that key inter-sentential phenomena, including reference resolution, temporal anchoring, and discourse relations, can be systematically encoded within the proposed framework, albeit with some limitations in cases requiring extra-textual knowledge or underspecified reference.

Finally, in terms of *annotation feasibility*, the observed disagreements between annotators highlight the main sources of difficulty, namely, omitted arguments, underspecified anaphora, and conjunctive adverbs’ semantic interpretation; while also suggesting that these issues can be resolved through clearer guidelines and annotation tool support.

Taken together, these results indicate that E-LMR may provide a viable basis for extending lexically grounded semantic representations to the discourse level while maintaining a balance between expressive power and annotation tractability.

5 Conclusion and Future Work

This paper has argued that inter-sentential relations constitute a necessary extension of semantic representation frameworks and that such an extension can be achieved without abandoning lexical anchoring. Building on the Lexicalized Meaning Representation (LMR) framework, we proposed Enhanced LMR (E-LMR) as a programmatic approach to capturing discourse-level phenomena, focusing on nominal and pronominal anaphora, temporal anchoring, and inter-sentential cohesion devices. E-LMR is conceived not as a purely manual annotation formalism, but as a modular semantic enrichment architecture supporting incremental automation.

The annotation experiment, conducted on a small corpus of European Portuguese narrative text, provides initial evidence that these relations can be systematically encoded while preserving the core principles of LMR, namely lexical transparency and structural interpretability. The results show that a significant proportion of semantic information, particularly argument realization, coreference, and discourse connectivity, emerges only when moving beyond the sentence level. At the same time, the analysis highlights the complexity of certain phenomena, such as lexical and underspecified anaphora, temporal anchoring across discourse spans, and the interpretation of connective elements, which often require contextual or world knowledge.

Rather than proposing a fully specified annotation scheme, this work advances a set of design principles and representation strategies that suggest the feasibility and potential of a lexically grounded approach to inter-sentential semantics. In contrast to more abstract, layered approaches such as UMR, E-LMR maintains a close alignment with surface realization, offering advantages in interpretability and annotation consistency, particularly in morphologically rich languages, such as Portuguese.

Several directions for future work follow naturally from this study. First, the development of a dedicated annotation interface is essential to support the semi-automatic enrichment process envisioned here, including modules for word-sense disambiguation, argument slot

instantiation, and discourse-level linking. Second, a more systematic evaluation is required, including the measurement of inter-annotator agreement and the comparison with alternative frameworks such as UMR. Third, the current treatment of temporal relations should be extended to incorporate tense, aspect, and modality, enabling a more complete model of temporal anchoring. Fourth, the representation of underspecified or extra-textual reference (e.g., **not-defined** nodes) calls for further theoretical and computational refinement, potentially integrating external knowledge sources.

Finally, the generality of the proposed approach should be tested across different genres and languages, in order to assess its robustness and adaptability. In this respect, E-LMR is best viewed not as a fixed formalism, but as a modular and extensible framework, contributing to a broader ecosystem of interoperable meaning representations. Large language models may also play a role in candidate generation for discourse relations, anaphora resolution, and underspecified reference interpretation. In this respect, E-LMR could provide an interpretable semantic layer for validating or constraining LLM-generated discourse inferences.

References

- 1 Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse (ACL 2013)*, pages 178–186, Sofia, 2013. ACL. URL: <https://aclanthology.org/W13-2322.pdf>.
- 2 Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract Meaning Representation (AMR) 1.2.6 Specification, 2019. URL: <https://github.com/amrisi/amr-guidelines/blob/master/amr.md>.
- 3 Jorge Baptista. Reconstruction of Zeroed Words in Lexicalized Meaning Representation. *Diacrítica*, 2026. Accepted for publication (in press).
- 4 Jorge Baptista and Nuno Mamede. *Dicionário Gramatical de Verbos do Português Europeu*. Universidade do Algarve, Faro, Portugal, novembro 2020. URL: <https://sapiencia.ualg.pt/handle/10400.1/15605>.
- 5 Jorge Baptista and Nuno Mamede. Syntactic Transformations in Rule-Based Parsing of Support Verb Constructions: Examples from European Portuguese. In Alberto Simões, Pedro Rangel Henriques, and Ricardo Queirós, editors, *9th Symposium on Languages, Applications and Technologies (SLATE 2020)*, volume 83 of *OpenAccess Series in Informatics (OASICS)*, pages 11:1–11:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/OASICS.SLATE.2020.11.
- 6 Jorge Baptista, Nuno Mamede, and Sónia Reis. Support Verb Constructions across the Ocean Sea. In *18th Workshop on Multiword Expressions @ LREC2022*, pages 26–36. ACL, 2022. URL: <https://hal.science/hal-03705534>.
- 7 Jorge Baptista, Izabela Müller, and Sónia Reis. Semantic Representation of Adverbs in the Lexicalized Meaning Representation (LMR) Framework. In Jorge Baptista and José Barateiro, editors, *14th Symposium on Languages, Applications and Technologies (SLATE 2025)*, volume 135 of *Open Access Series in Informatics (OASICS)*, pages 9:1–9:18, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/OASICS.SLATE.2025.9.
- 8 Jorge Baptista, Simone Pereira, and Nuno Mamede. ZAC – Zero Anaphora Corpus: A Corpus for Zero Anaphora Resolution in Portuguese. In Hilário Leal Fontes and António Branco, editors, *Proceedings of the Workshop on Corpora and Tools for Processing Corpora, (co-located with 12th International Conference Computational Processing of the Portuguese Language – PROPOR’2016)*, pages pp. 38–45. PROPOR 2016/Portuguese Language Department, Directorate-General for Translation, European Union, 2016.

- 9 Jorge Baptista, Sónia Reis, João Dias, and Pedro A. Santos. Lexicalized Meaning Representation (LMR). In Claire Bonial, Julia Bonn, and Jena D. Hwang, editors, *Proceedings of the Fifth International Workshop on Designing Meaning Representations (DMR) @ LREC-COLING 2024*, pages 101–111, Torino, Italy, May 2024. ELRA and ICCL. doi:10.34626/6080.
- 10 Jorge Baptista, Sónia Reis, Pedro A. Santos, and João Dias. Comparação de anotações AMR entre línguas: problemas e soluções. *Revista da Associação Portuguesa de Linguística*, 1(11):27–54, 2024. doi:10.26334/2183-9077/rapln11ano2024a2.
- 11 Julia Bonn, Claire Bonial, Matt Buchholz, Hsiao-Jung Cheng, Alvin Chen, Ching-wen Chen, Andrew Cowell, William Croft, Lukas Denk, Ahmed Elsayed, et al. Uniform Meaning Representation 2.0, 2025. LINDAT/CLARIAH-CZ Digital Library.
- 12 Julia Bonn, Matthew J. Buchholz, Jayeol Chun, Andrew Cowell, William Croft, Lukas Denk, Sijia Ge, Jan Hajič, Kenneth Lai, James H. Martin, et al. Building a Broad Infrastructure for Uniform Meaning Representations. In *Proceedings of LREC-COLING*, pages 2537–2547, 2024.
- 13 Julia Bonn, Ching-wen Chen, Andrew Cowell, William Croft, Lukas Denk, Jan Hajič, Kenneth Lai, Martha Palmer, Alexis Palmer, James Pustejovsky, et al. Uniform Meaning Representation, 2023. LINDAT/CLARIAH-CZ Digital Library.
- 14 Julia Bonn, Skatje Myers, Jens E. L. Van Gysel, Lukas Denk, Meagan Vigus, Jin Zhao, Andrew Cowell, William Croft, Jan Hajič, James H. Martin, Alexis Palmer, Martha Palmer, James Pustejovsky, Zdenka Urešová, Rosa Vallejos, and Nianwen Xue. Mapping AMR to UMR: Resources for adapting existing corpora for cross-lingual compatibility. In Daniel Dakota, Kilian Evang, Sandra Kübler, and Lori Levin, editors, *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 74–95, Washington, D.C., 2023. Association for Computational Linguistics. URL: <https://aclanthology.org/2023.tlt-1.8/>.
- 15 Jayeol Chun and Nianwen Xue. Uniform meaning representation parsing as a pipelined approach. In *Proceedings of TextGraphs*, pages 40–52, 2024.
- 16 Michael Wayne Goodman. Penman: An open-source library and tool for AMR graphs. In Asli Celikyilmaz and Tsung-Hsien Wen, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 312–319, Online, 2020. Association for Computational Linguistics. doi:10.18653/v1/2020.acl-demos.35.
- 17 Caroline Hagège, Jorge Baptista, and Nuno Mamede. Caracterização e Processamento de Expressões Temporais em Português. *Linguamática*, 2(1):63–76, 2010. URL: <https://www.linguamatica.com/index.php/linguamatica/article/download/47/60/>.
- 18 M. A. K. Halliday and Ruqaiya Hasan. *Cohesion in English*. Longman, London, 1976.
- 19 M. A. K. Halliday and Christian M. I. M. Matthiessen. *Halliday's Introduction to Functional Grammar*. Routledge, London, 4 edition, 2014.
- 20 Zellig S. Harris. Discourse Analysis. *Language*, 28(1):1–30, 1952.
- 21 Erhard Hinrichs. Temporal Anaphora in Discourses of English. *Linguistics and Philosophy*, pages 63–82, 1986.
- 22 Behrooz Mansouri. Survey of Abstract Meaning Representation: Then, Now, Future, 2025. arXiv preprint. doi:10.48550/arXiv.2505.03229.
- 23 João Silvestre Marques. Anaphora resolution in portuguese: A hybrid approach. Master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, November 2013. MSc dissertation.
- 24 Christian Molinier and Françoise Lévrier. *Grammaire des Adverbes: Description des Formes en -ment*. Droz, Genève, 2000. URL: https://books.google.com/books/about/Grammaire_des_adverbes.html?id=HSHMkiPy4pgC.
- 25 I. Müller, N. Mamede, and Jorge Baptista. Adverbios conjuntivos multipalabra en el portugués brasileño: Un enfoque metodológico basado en el léxico-gramática. *Studia Linguistica Romanica*, 2026. Accepted for publication (in press).

- 26 Simone Pereira. ZAC.PB: An Annotated Corpus for Zero Anaphora Resolution in Portuguese. In *Student Research Workshop in conjunction with RANLP-09*, pages 53–59, September 2009. URL: <https://aclanthology.org/R09-2010/>.
- 27 Nathan Schneider, Jeff Flanigan, and Tim O’Gorman. The Logic of AMR. In *Proceedings of the 2015 Conference of the Association for Computational Linguistics (ACL 2015)*, pages 4–5, Beijing, 2015. ACL. doi:10.3115/v1/N15-4003.
- 28 Jens Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O’Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. Designing a Uniform Meaning Representation for Natural Language Processing. *KI – Künstliche Intelligenz*, 35(3):343–360, 2021. doi:10.1007/S13218-021-00722-W.
- 29 Jiarui Yao, Haoling Qiu, Bonan Min, and Nianwen Xue. Annotating Temporal Dependency Graphs via Crowdsourcing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5368–5380, 2020. doi:10.18653/v1/2020.emnlp-main.434.
- 30 Yuchen Zhang and Nianwen Xue. Neural ranking models for temporal dependency structure parsing. *arXiv preprint*, 2018. arXiv:1809.00370.
- 31 Yuchen Zhang and Nianwen Xue. Structured Interpretation of Temporal Relations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3222–3229, Miyazaki, Japan, 2018. European Language Resources Association (ELRA). URL: <https://aclanthology.org/L18-1490>.
- 32 Jin Zhao, Nianwen Xue, Jens E. L. Van Gysel, and Jinho D. Choi. UMR-Writer: A Web Application for Annotating Uniform Meaning Representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 160–167, Online, 2021. Association for Computational Linguistics. doi:10.18653/v1/2021.emnlp-demo.19.

A Topic–Word Graph for Topic Modeling Visualization

Ana Rita Peixoto ✉ 

ISTAR, Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

Abstract

Topic Modeling (TM) is a widely used method for discovering hidden themes in unlabeled text collections, but the resulting topics are often difficult to interpret. Most existing visualizations present topics separately and do not show the words that link them, offering only a partial view of how topics relate to one another. This work introduces a graph-based representation in which topics and words are modeled as connected nodes. This structure makes shared words explicit and allows users to observe how individual terms contribute to multiple thematic areas within the corpus. By encoding topic–word relationships as edges, the visualization reveals the distribution of these connections and provides a more detailed perspective than traditional topic-centric displays. The graph includes four node types, defined by their role and degree (number of connections). The developed tool, `topicwordgraph`, offers two complementary visualizations: a main graph representation that distinguishes node types by color, and two bar charts that summarize topic overlap and word memberships. Together, these views help users understand how topics intertwine through shared vocabulary. By making the shared words between topics explicit, this tool offers a practical and effective way to interpret how topics are interconnected.

2012 ACM Subject Classification Computing methodologies → Natural language processing

Keywords and phrases Topic Modeling, Data Visualization, Graph-Based Representation, Topic Interpretability, Text Mining

Digital Object Identifier 10.4230/OASICS.SLATE.2026.6

Supplementary Material *Software (Source Code)*: <https://github.com/ritapx/topicwordgraph> [14]; archived at `swb:1:dir:2f09eb9f388de51f82caa9f5e7a78d537a07314a`

Funding This work was partially supported by national funds through Fundação para a Ciência e a Tecnologia, I.P. (FCT), ISTAR-Iscte Pluriannual Projects UID/04466/2025 (<https://doi.org/10.54499/UID/04466/2025>).

1 Introduction

When there is a need to discover themes within text corpora, independent of the research domain, the primary technique used is Topic Modeling (TM) [3]. In this framework, the model outputs a list of the top- k words associated with each topic, along with the corresponding topic-word distribution. This list is used both to describe the topics and to evaluate the model’s overall performance and interpretability. However, top- k word lists may not always provide sufficient semantic clarity, as they can hide topic coherence and delay human interpretability [9]. As a result, newer studies have emphasized the need for complementary evaluation strategies that better capture the nuanced structure of topic–word relationships [1], highlighting how central these connections are for understanding what each topic truly represents and how they can be connected.

Despite their computational simplicity, topic models often present interpretational and communicational challenges when reduced to word lists and word-cloud visualizations [5, 11]. Some visualizations have been developed to support understanding of topic models and to assess whether topics are well separated, such as the well-known LDAvis tool [17] applied to the Latent Dirichlet Allocation (LDA) algorithm [4]. Nevertheless, these approaches tend



© Ana Rita Peixoto;

licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 6; pp. 6:1–6:11

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

to be more effective for experts in the field than for communicating results to a broader audience [3]. Highlighting the need to develop complementary visualization strategies that bridge that gap and capture the topic-word connections.

The visualization of topics and their connecting words remains largely unexplored. Most existing approaches focus on isolated topic representations rather than on shared vocabulary. Although an ideal topic model would exhibit no overlap in their top words, this assumption rarely holds in practice, where topic boundaries tend to blur, and word overlap naturally emerges [1]. In practice, word overlap across topics reflects the inherent interconnections among them and often reveals a central theme that summarizes the source corpus. With this work, we aim to address this gap by answering the following research question:

- How can topic models be visualized to effectively reveal and interpret the interconnection of topics through shared words?

To address this research question, a graph-based visualization was developed and implemented in Python - `topicwordgraph`¹. The objective of this tool is to provide a visual aid for how topics intertwine, to uncover the specific words that connect them, and to provide a clearer representation of the thematic and semantic structure underlying the data.

2 Related Work

Identifying meaningful patterns and extracting themes is a common need across various domains, and TM can perform it effectively [3]. Recent studies use it for various tasks, from a complementary tool in systematic literature reviews [7] to its application in applied genomic science [20].

Topic Modeling Visualization

Most studies, when interpreting topics, use word clouds and tables or bar charts with the top words for each topic. The following is a brief description of some TM visualization tools that aim to improve that examination.

LDavis is a combined visualization that aims to increase the interpretability of each topic, understand how prevalent each topic is, and how topics relate to each other [17]. For that, it offers two visualizations: a global view of the topic model and a bar chart showing the words most useful for interpreting the selected topic. The global view of the topic model is a 2D plane, with topics represented as circles; the distance between circles indicates similarity, and their sizes indicate prevalence.

A recent study proposes an integrated application called WordMap [18]. The authors explain that most tools are segmented, offering a single tool with various text mining techniques. In terms of visualization, this tool provides: word clouds, word distributions over topics in a bar chart, an inter-topic distance map (equivalent to LDavis), topic distributions over the corpus, a hierarchical clustering graphic with semantic associations, and a heat map for visualizing semantic similarity.

Similar to the WordMap, the `topicwizard` was designed to enhance topic visualization [11]. It also incorporates an inter-topic distance map, topic-word distributions, and word clouds. However, it differs by presenting a word map that enables interactive analysis of the corpus's semantic landscape using word embeddings, and by providing a document view exploratory analysis.

¹ <https://github.com/ritapx/topicwordgraph>

However, these tools are mostly used for the interpretation and evaluation of topics by field experts. To the best of our knowledge, there is a gap in the communication of the extracted topics. As well as a gap in the analysis of the shared words by topic as a semantic connection between the topics, and as a description of the data domain used in the TM.

3 Topic-Word Graph

A Python tool was created for TM visualization, the `topicwordgraph`, enabling the extraction of information about the words that define each topic and their interconnections. It transforms a topic model into a network with the structure of the Topic-Word graph.

This section begins by formally specifying the Topic-Word graph using the TM components, the topics (T), and the words (W). Then it explains how to interpret a network obtained by applying a specific model to the Topic-Word graph. Thirdly, it describes how visualization elements are utilized to facilitate the interpretation of the resulting graph. Finally, it briefly explains the implementation and how to use the developed tool.

3.1 The Graph

TM produces a collection of n topics, which we denote by $T = \{T_1, T_2, \dots, T_n\}$. Each topic T_i is represented by its top- k words, $T_i = \{w_1, w_2, \dots, w_k\}$, these are the words most strongly associated with the topic according to the model. By taking the union of all topic words, we obtain the set W , which contains every distinct word that appears in at least one topic: $W = \bigcup_{i=1}^n T_i$. This set represents the vocabulary used to describe the topics.

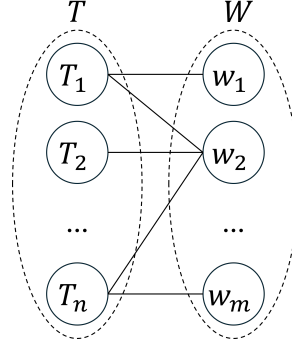
A graph is a mathematical structure defined as a pair $G = (V, E)$, where V is a set of nodes and $E \subseteq V \times V$ is a set of edges that represent relationships between nodes [6]. To define a topic model as a graph, we represent the nodes as topics or words, and the model's relationships as edges, specifically, the membership of a word in a topic.

The node set is $V = T \cup W$, meaning that every topic $T_i \in T$ and every word $w \in W$ is a node. These two types of nodes form two disjoint subsets. The set of edges, E , is the set of connections between words and topics. Each edge means that a word belongs to a topic, resulting in a bipartite graph where $E \subseteq T \times W$. No edges occur between two topics or between two words. This restriction creates a bipartite graph, with topics on one side and words on the other. The edges represent the relation “word belongs to topic,” which does not require directionality for analysis. Additionally, the edges do not need a weight assigned since they represent a single membership.

The Topic-Word graph, G , is therefore an undirected, unweighted, and bipartite graph that can be represented as follows: $G = (T \cup W, E)$, $E = \{(T_i, w) | w \in T_i\}$. Providing a structural representation of the topic model, where topics and words are nodes, and edges encode the membership relation between them. Figure 1 illustrates the Topic-Word graph.

3.2 The Interpretation

Once defined, the Topic-Word Graph enables the use of graph-theoretic characteristics to analyze its structure and meaning. These will provide valuable insight into how topics relate to one another, how words are distributed across topics, and how the vocabulary contributes to the overall coherence of the topic model. Allowing the extraction of new information that would otherwise be harder to conclude.



■ **Figure 1** Topic-Word Graph illustration, where T is the set of the n topics in the topic model, W the union of the top- k words of each topic (resulting in m different words) and $G = (T \cup W, E)$, $E = \{(T_i, w) | w \in T_i\}$, a undirected, unweighted and bipartite graph ($E \subseteq T \times W$).

The Topic–Word Graph naturally facilitates the analysis of words that appear across multiple topics. The graph measure employed to know which word belongs to one or more topics is the node degree. The degree of a node is the sum of the edges linked to it [2]. Since it is a bipartite graph, there are no word nodes linked to other word nodes, so the degree value indicates the number of topics the word belongs to.

The degree of a word node can vary from one to n , the number of topics. When the degree of a word is one, it means that the word only belongs to a topic, and when it is n , it belongs to all the topics. Additionally, the degree of all topic nodes is k , the number of top words used to create the graph. For instance, in Figure 1, the degree of the word node w_1 is one, indicating that it belongs to only one topic. And the degree of the word w_2 is three, belonging to three topics. We define a word node with a degree of one, a *singleton word*, because it belongs to only one topic. In contrast, a *central word* is a word node with a degree value greater than one, meaning it belongs to more than one topic. When a *central word* has degree n , it belongs to all topics and is a special case; it is called a *global central word*.

The *singleton words* (degree = 1) appear exclusively in a single topic and therefore contribute uniquely to the semantic interpretation of that topic. Since *singleton words* do not connect across topics, they play no role in inter-topic relations and instead work as a reinforcement of the distinctiveness of individual topics. In a coherent topic model, each topic should be unique and specific [13].

In contrast, *central words* (degree > 1) act as connectors between topics. Understanding *central words* is important because they reveal the shared theme across topics. When those words appear in all the topics, with a degree value of n , the *global central words* describe and summarize the data. Furthermore, if a model has mostly *central words*, they can reveal semantic overlap between the topics, suggesting that those topics may not be the most suitable [13, 16].

3.3 The Visualization

The graph visualization incorporates visual elements to enable the interpretation explained. In terms of format, circles represent the graph’s nodes, and in the middle is written the name of each topic or the specific word. Topic nodes are larger than word nodes to help separate the two concepts. Lastly, to draw the nodes in a dispersed, non-overlapping layout, the `networkx` library’s `spring_layout` was used. Additionally, to allow the user to change

the layout as desired and improve the communication of results, the graph is interactive, enabling them to move the nodes. The visualizations were implemented using two Python libraries: `networkx` [8] and `matplotlib` [10].

Color is the key element in this visualization, enabling immediate understanding of the four node types: *topics*, *singleton words*, *central words*, and *global central words*. We defined a default palette, the one applied in the visualizations of the results section above. The default palette is: blue for the *topics* and a lighter blue for the *singleton nodes*, since they characterize the topics; and green for the *central words*, with a darker shade of green for the *global central words* to create an emphasis. We choose this approach to achieve a comprehensive analysis.

Lastly, the `topicwordgraph` tool allows the creation of two bar charts as a complementary visualization of the graph. A bar chart named “Topics with most overlapping words”, which shows the count of the overlapping words for each topic name. Enabling easy recognition of the topics with higher or lower levels of interconnection. Additionally, a second bar chart named “Words with degree > 1”, which displays the *central words* and their degree values. Facilitating the distinction between them, effortlessly identifying which ones are the ones that create the interconnection between the topics.

3.4 The Tool Usage

To use the `topicwordgraph` tool, the user needs to provide the topic names and the corresponding top-k words in a dictionary format where the key is the topic name, and the value is a list of words. The following is a simple example of how to use the tool to create a graph visualization (`.plot()`) and the bar charts (`.plot_bar_charts()`).

```
from topicwordgraph import TopicWordGraph

topics = {
    "Topic Name 1": ["word1", "word2", "word3"],
    "Topic Name 2": ["word1", "word4", "word5"],
    "Topic Name 3": ["word2", "word6", "word7"]
}

twg = TopicWordGraph.from_named_topics(topics)

# Interactive Topic-Word graph
twg.plot()

# Bar charts
twg.plot_bar_charts()
```

4 Results

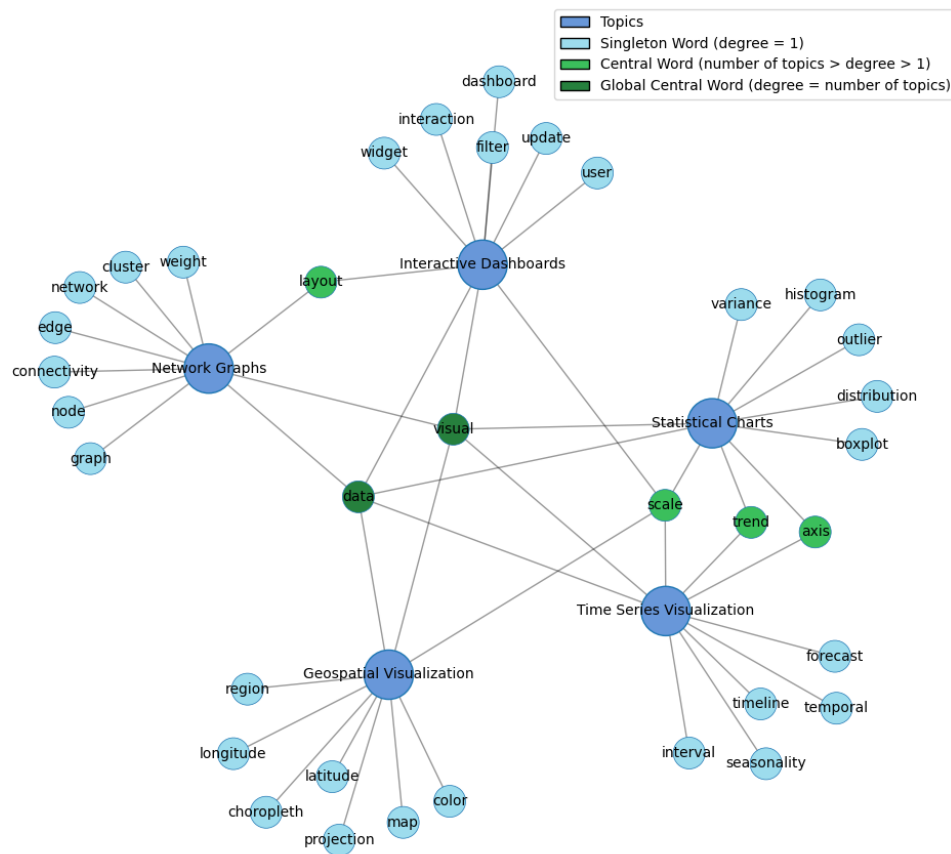
The purpose of this Section is to showcase visualizations created with the `topicwordgraph` tool, emphasizing its advantages and limitations. For that, three different TM were used to create the visualizations: one from a prompt in a large language model, the GPT-5.3 [12]; another from the results of a literature paper that applied a TM to startups’ tweets [15]; and, lastly, one from a classic example with the data of the 20 newsgroups dataset [19].

Simulation with GPT-5.3

It was important to create a visualization that illustrates all the components, including the four node types that enable the understanding of the interconnections in a topic model. Since most real-world applications may not allow us to achieve that goal, we simulate a topic model using a prompt in ChatGPT with the GPT-5.3 version. In the prompt, it was assured that all node types would appear in the result.

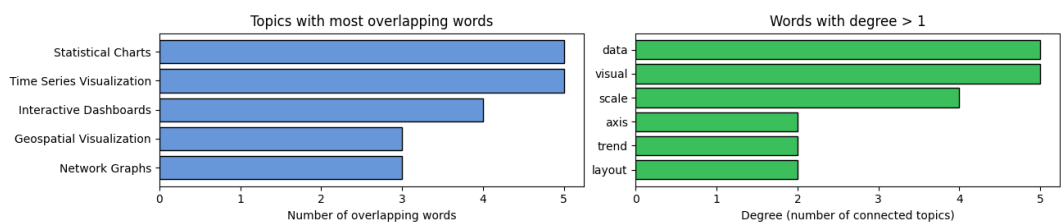
The following is the prompt used: *Please simulate a topic modeling and provide a list of five topics and their top 10 words. One or two words must belong to the five topics; other words must belong to more than one topic, but not all topics. You must have words that are common to more than one topic, and words that belong only to one topic. The topic model result must be a good model, and the domain should be data visualization. The format of the result must be: “topics = { “Topic name 1”: [“word1”, “word2”, ...], “Topic name 2”: [“word3”, “word 4”, ...] }”.*

The resultant list was then applied to the graph to create the visualizations. Figure 2 shows the graph, and the corresponding bar charts are in Figure 3.



■ **Figure 2** Topic-word graph visualization of the topic model created by ChatGPT (GPT-5.3) about data visualization.

The graph (Figure 2) shows the five topics: “Interactive Dashboards”, “Network Graphs”, “Geospatial Visualization”, “Statistical Charts”, and “Time Series Visualization”. The topics “Statistical Charts” and “Time Series Visualization” have the highest word overlap. The



■ **Figure 3** Bar charts of the topic model created by ChatGPT (GPT-5.3) about data visualization.

topic “Interactive Dashboards” is connected to “Network Graphs” via the word “layout” and to “Statistical Charts” via the word “scale”. The visualization shows that the words “data” and “visualization” are central to the model’s domain, connecting all the topics and revealing the shared theme across them. The bar charts, in Figure 3, complement those findings.

From the visualizations, it is possible to extract that the model is about data visualization, using the *global central words* “data” and “visual”, further “layout”, “scale”, “trend”, and “axis” are valuable concepts in that domain, since they connect other topics.

Topics of Startups’ Tweets

The following is a scenario of TM applied to real data. The authors in [15] used TM to understand what Portuguese IT startups posted on Twitter from 2015 to 2020, using a dataset of 15,557 tweets from eight startups. The results comprise a coherent model (using LDA) with five topics described by their 10 most important words, which enabled the application of the results to a Topic-Word graph.

One of the conclusions was that startups posted about the following five topics: “Fintech and ML”, “IT”, “Bank and funding”, “Business operations”, and “Product / service RD”. Additionally, the authors performed a time-based analysis, discovering that those themes predominance change along the startup’s maturity. These are valuable findings, but applying graph visualization to the results may enable us to extract more knowledge. Figure 4 shows the resultant graph, and the corresponding bar charts are in Figure 5.

From the visualizations, it is possible to see that the words “lisbon”, “learn”, “webinar”, “machinelearning”, “cloud”, “service”, and “read” are *central words*. Together, these words characterize the domain of Portuguese IT startups, highlight recurring themes in their content, and reveal what these startups always focus on. This analysis could enrich the work already done.

Topics of the 20 Newsgroup Dataset

The 20 newsgroup dataset is widely used and comprises approximately 20,000 newsgroup documents. In [19], the authors applied various TM algorithms to the dataset and evaluated the results using annotators. The evaluation was made based on the top-10 most important words for the topics. The MRF-LDA achieved the best model with four topics: “Crime”, “Sex”, “Sports”, and “Health”. The list of top words and the topic names enabled the use of the `topicwordgraph` tool. The resulting visualization of applying the top-10 words for each topic to the Topic-Word graph is in Figure 6, and Figure 7 displays the complementary bar chart.

The graph visualization provides an easy overview of the four topics: two are completely independent, with only *singleton words*, and one *central word* connects the other two. Since only one of the 40 words belongs to more than one topic, this implies that those four topics

6:8 A Topic–Word Graph for Topic Modeling Visualization

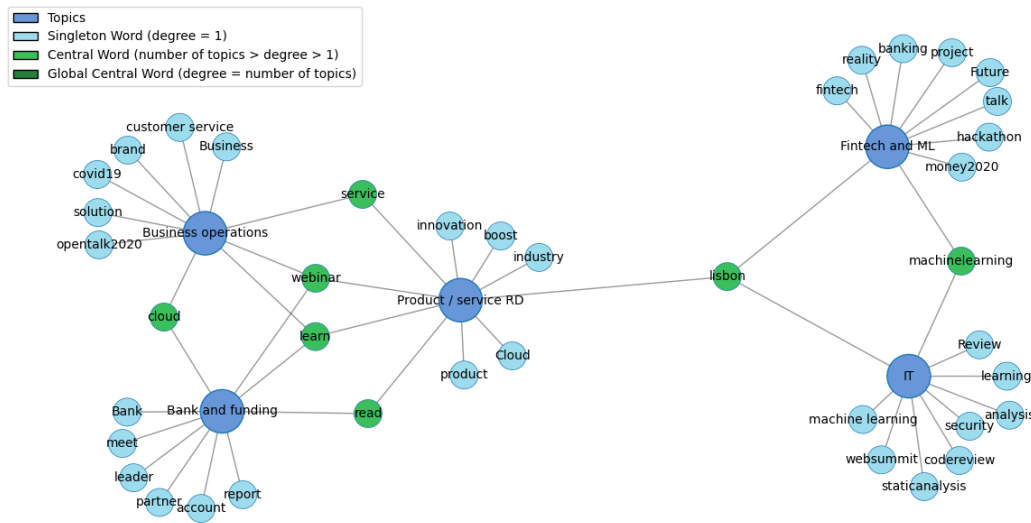


Figure 4 Topic-word graph visualization of the topic model about what Portuguese IT startups posted on Twitter [15].

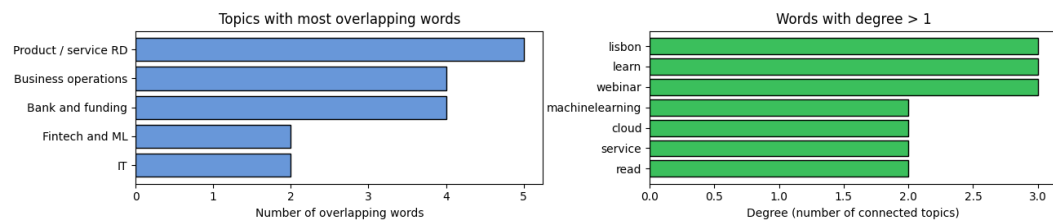


Figure 5 Bar charts of the topic model about what Portuguese IT startups posted on Twitter [15].

are very well separated. The *central word* is “child”, connecting the topics “Crime” and “Sex”, suggesting that some news combine those two topics with children involved. This analysis, enabled by the visualization, reveals how the newsgroup topics can be interconnected and allows a clear visualization of theme separations across the other topics.

5 Conclusions

The TM representation as a Topic–Word graph, using the developed tool, enables the visualization of: 1) how separated the topics are; 2) how some *central words* thematically connect some topics; and 3) how *central words* characterize and summarize the data domain. These three key points outline a comprehensive analysis of the TM resulting models. By representing topics and their shared words as a connected network, based on the Topic–Word graph, it is possible to effectively reveal and interpret the interconnections among topics, directly answering the proposed research question.

The main purpose of this tool’s visualizations is to communicate TM results, complementing more typical visualizations, such as word clouds, with topic-word connections. However, it can also be applied to evaluate whether topics are sufficiently separated and, at the same time, whether the extracted topics well represent the data. Allowing it to be an intermediate visualization while fine-tuning a model or to provide to annotators.

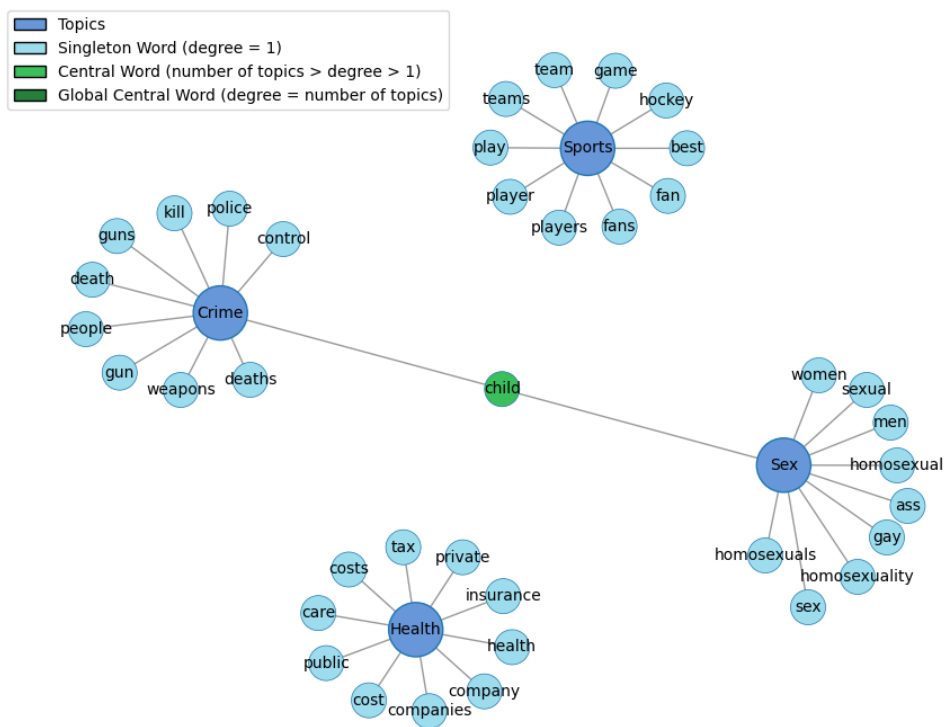


Figure 6 Topic-word graph visualization of the topic model applied to the 20 newsgroup dataset [19].

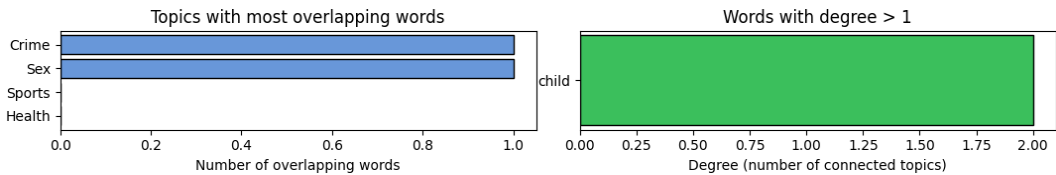


Figure 7 Bar charts of the topic model applied to the 20 newsgroup dataset [19].

Limitations

The proposed graph-based visualization has some limitations. One of them is scalability and size, since topic models with more topics produce more words, increasing visual complexity. Secondly, this work assumes that shared words imply semantic connections between topics. However, some words may appear across topics due to their high frequency, even after suitable preprocessing and the removal of stopwords. Lastly, it does not allow hierarchical connections; the graph shows the relation between topic and word, and does not include relationships with subtopics. Restricting the visualization to non-hierarchical topic models.

Future Work

Several directions can extend and strengthen this work. First, future research could focus on improving the scalability of the visualization by incorporating some filtering, such as edge-reduction techniques, to prevent visual clutter and maintain interpretability. Second, the exploration of more sophisticated methods for identifying meaningful shared words, as the

current work focuses only on degree values. Third, enable hierarchy among topics, expanding the Topic–Word graph to include a new type of node: subtopics. Lastly, another promising direction is to incorporate TM evaluation metrics directly into the visualization, enabling effortless assessment of model quality.

References

- 1 Ana Amaro and Fernando Bacao. Topic modeling: A consistent framework for comparative studies. *Emerging Science Journal*, 8:125–139, February 2024. doi:10.28991/ESJ-2024-08-01-09.
- 2 Albert-László Barabási. *Network Science*. Cambridge University Press, 2016. URL: <http://networksciencebook.com/>.
- 3 Monika Bednarek. Topic modelling in corpus-based discourse analysis: Uses and critiques. *Discourse Studies*, 27:659–671, August 2025. doi:10.1177/14614456241293075.
- 4 David M Blei, Andrew Y Ng, and Jordan@cs Berkeley Edu. Latent dirichlet allocation michael i. jordan. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- 5 Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. Reading tea leaves: how humans interpret topic models. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, NIPS’09, pages 288–296, Red Hook, NY, USA, 2009. Curran Associates Inc. URL: <https://proceedings.neurips.cc/paper/2009/hash/f92586a25bb3145facd64ab20fd554ff-Abstract.html>.
- 6 Reinhard Diestel. *Graph Theory*. Springer, 5 edition, 2017. doi:10.1007/978-3-662-53622-3.
- 7 Priyanka Gupta, Bosheng Ding, Chong Guan, and Ding Ding. Generative ai: A systematic review using topic modelling techniques. *Data and Information Management*, 8, June 2024. doi:10.1016/j.dim.2024.100066.
- 8 Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, 2008.
- 9 Alexander Hoyle, Pranav Goel, Denis Peskov, Andrew Hian-Cheong, Jordan Boyd-Graber, and Philip Resnik. Is automated topic model evaluation broken? the incoherence of coherence. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS ’21, Red Hook, NY, USA, 2021. Curran Associates Inc.
- 10 J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi:10.1109/MCSE.2007.55.
- 11 Márton Kardos, Kenneth C. Enevoldsen, and Kristoffer Laigaard Nielbo. topicwizard – a modern, model-agnostic framework for topic model visualization and interpretation. In *Proceedings of the 8th International Conference on Natural Language and Speech Processing (ICNLSP-2025)*, May 2025. doi:10.48550/arXiv.2505.13034.
- 12 OpenAI. GPT-5.3 (ChatGPT). <https://chat.openai.com/>, 2026. Large language model.
- 13 Derek O’Callaghan, Derek Greene, Joe Carthy, and Pádraig Cunningham. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*, 42(13):5645–5657, 2015. doi:10.1016/j.eswa.2015.02.055.
- 14 Ana Rita Peixoto. topicwordgraph. Software, version 1.0., swhId: swh:1:dir:2f09eb9f388de51f82caa9f5e7a78d537a07314a (visited on 2026-07-17). URL: <https://github.com/ritapx/topicwordgraph>, doi:10.4230/artifacts.27313.
- 15 Ana Rita Peixoto, Ana de Almeida, Nuno António, Fernando Batista, and Ricardo Ribeiro. Diachronic profile of startup companies through social media. *Social Network Analysis and Mining*, 13:1–18, 2023. doi:10.1007/s13278-023-01055-2.
- 16 Emil Rijcken, Kalliopi Zervanou, Pablo Mosteiro, Floortje Scheepers, Marco Spruit, and Uzay Kaymak. Topic specificity: A descriptive metric for algorithm selection and finding the right number of topics. *Natural Language Processing Journal*, 8:100082, 2024. doi:10.1016/j.nlp.2024.100082.

- 17 Carson Sievert and Kenneth E Shirley. Ldavis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, 2014.
- 18 Zhijian Wei, Huiwen Zou, Patrick Cheong Iao Pang, Penny Wong On Chao, and Benjamin K. Ng. Wordmap: Text mining application of enhanced corpus segmentation and semantic topic recognition. *Applied Sciences (Switzerland)*, 15, June 2025. doi:10.3390/app15126632.
- 19 Pengtao Xie, Diyi Yang, and Eric Xing. Incorporating word correlation knowledge into topic modeling. In Rada Mihalcea, Joyce Chai, and Anoop Sarkar, editors, *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 725–734, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi:10.3115/v1/N15-1074.
- 20 Yidong Zhang, Xilin Jiang, Alexander J. Mentzer, Gil McVean, and Gerton Lunter. Topic modeling identifies novel genetic loci associated with multimorbidities in uk biobank. *Cell Genomics*, 3, August 2023. doi:10.1016/j.xgen.2023.100371.

Polarizations in Static Word Embeddings: Investigating Bias in Marginalized Dialects of Portuguese

Raimundo Juracy Campos Ferro Junior ✉ 

Universidade Federal do Ceará, Fortaleza, Brasil

ISTAR, Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

Eugénio Ribeiro ✉ 

INESC-ID Lisboa, Portugal

ISTAR, Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

Leonardo Sampaio Rocha ✉ 

Universidade Federal do Ceará, Fortaleza, Brasil

Abstract

This study investigates bias in static word embeddings applied to Pajubá, a dialect spoken within the Brazilian LGBTQIAP+ community. Four models from the NILC repository (Word2Vec, GloVe, FastText, and Wang2Vec) were evaluated across two dimensions: representational capacity and polarity analysis. The vocabulary coverage test revealed that approximately 74% of dialectal terms are present in the models' vector spaces. The RND and SC-WEAT tests consistently point toward negative associations for identity terms across all models, with the SC-WEAT further suggesting that reappropriated dialect terms carry their standard Portuguese positive valence into the embedding space, while explicit identitarian markers remain negatively encoded. An adjectivation test confirms stereotypical associations, including the linkage of *travesti* with criminality and the pathologisation of *gay*. These findings suggest that static word embeddings fail to capture the sociolinguistic complexity of Pajubá and systematically reflect the dominant and often discriminatory discourses present in the training corpora.

2012 ACM Subject Classification Computing methodologies → Natural language processing

Keywords and phrases word embeddings, bias detection, Portuguese, LGBTQIAP+, pajubá, NLP, lexical association, static embeddings

Digital Object Identifier 10.4230/OASICS.SLATE.2026.7

Funding This work was supported by Portuguese national funds through Fundação para a Ciência e a Tecnologia, I.P. (FCT) under projects UID/50021/2025 (DOI: <https://doi.org/10.54499/UID/50021/2025>), UID/PRR/50021/2025 (DOI: <https://doi.org/10.54499/UID/PRR/50021/2025>), and UID/04466/2025 (DOI: <https://doi.org/10.54499/UID/04466/2025>).

Raimundo Juracy Campos Ferro Junior: Funded by CAPES (PDSE scholarship)

1 Introduction

Text embedding models constitute the backbone of modern Natural Language Processing (NLP) systems. These algorithms transform linguistic units (texts, documents, sentences, expressions, and words) into continuous numerical vector representations that capture the semantic and syntactic relations of the source language.

Such embedding techniques enable algorithms, which are otherwise limited to numerical inputs, to interpret contextual similarities through mathematical operations. Although efficient for large-scale applications and widely adopted across diverse contexts, these models inherit structural patterns from their training corpora, encoding not only linguistic information but also latent **sociocultural biases** present in the data [4].



© Raimundo Juracy Campos Ferro Junior, Eugénio Ribeiro, and Leonardo Sampaio Rocha; licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 7; pp. 7:1–7:15

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Investigating these biases reveals that discriminatory associations, such as gender stereotypes (e.g., *nurse* linked to women) and the negative racialization of certain terms, are systematically amplified in downstream applications (e.g., résumé screening, credit scoring). Several studies have sought to mitigate the effects of such biases [18, 15, 11, 9].

However, much of the research on bias in embedding models has focused on resources and techniques validated for English, rarely extending to Portuguese. There is therefore a significant gap in the literature regarding the evaluation of embeddings for Portuguese particularly across its dialectal and sociocultural varieties with a lack of analyses that account for the lexical and contextual specificities of the language.

This work addresses that gap. The primary objective is to evaluate the ability of embedding models to consistently represent the semantics of dialects belonging to marginalized LGBTQIAP+ communities. We hypothesize that, due to the limited presence of these dialects in widely used training corpora, terms characteristic of such linguistic varieties tend to be inadequately or entirely absent from the resulting embeddings.

We further investigate whether models can capture the contextual meanings of dialect words, both those shared with standard Portuguese and those exclusive to the dialect, as employed by their speakers. A salient example is the term *travesti* in the dialect Pajubá, where it functions as a neutral identity marker (carrying neither positive nor negative connotation) to address trans women, yet may be associated with negative connotations by the models due to biases present in the training data.

The specific objectives of this study are therefore: (1) to determine whether embedding models can generate vector representations for texts composed predominantly of marginalized dialect terms; and (2) to analyze whether such models reflect semantic biases by associating LGBTQIAP+ identity terms with negative or positive polarities through semantic polarity analysis.

2 Related Work

Bias detection in word embeddings has been an active research area since static embeddings were shown to encode stereotypical occupational associations [4], and WEAT was established as a standard intrinsic bias measurement tool [5]. Despite the breadth of subsequent work, studies remain concentrated on English and on gender as the primary dimension of analysis, with sexual orientation, non-binary gender identities, and intersectional bias receiving considerably less attention [23].

The present work differs from this mainstream in two fundamental ways. First, it investigates a dimension of bias that remains underexplored: the association of LGBTQIAP+ vocabulary with negative attributes in static word embeddings. Second, it operates on a minority linguistic variety within Portuguese, Pajubá, rather than on standard register vocabulary.

Among the few studies addressing sexual orientation bias, [21] applied a WEAT-inspired metric using *straight* versus *gay* as target poles in German embeddings, finding that homosexuality-related terms clustered near concepts of violence, prohibition, and abuse, while heterosexuality was associated with positive sentiment. Our results for Portuguese show a similar directional pattern. However, their binary straight/gay opposition is insufficient for the broader spectrum of identities investigated here, which motivates the adoption of SC-WEAT in this work. [8] further showed, across fifteen English embedding models, that static embeddings systematically associate marginalised identities including LGBTQ groups with offensive vocabulary.

On the methodological side, [24] demonstrated that grammatical gender acts as a confounding signal in bias measurements for gendered languages, motivating the morphological adaptations made to the SC-WEAT attribute sets in this work. [7] provide the most directly comparable precedent, applying SC-WEAT to Portuguese GloVe embeddings and confirming that English bias metrics can be adapted to Portuguese with appropriate morphological adjustments.

3 Methodology

To achieve the proposed objectives, we adopted an approach grounded in tests consolidated in the literature. The investigation focused on a case study of Pajubá, a crypto-dialect developed and used by Brazilian LGBTQIAP+ communities, particularly by trans women. Further details on the dialect and the data collection process are provided in Section 3.1.

The experiments employed embedding models available in the **NILC** (Núcleo Inter-institucional de Linguística Computacional) word embedding repository [14], covering four techniques: Word2Vec [19], GloVe [22], FastText [3], and Wang2Vec [17]. Details regarding the training corpus and model parameters are discussed in Section 3.3.

To address Objective 1 (representational capacity), we applied a **vocabulary coverage test**, which quantifies the proportion of dialect terms present in the models' vector space.

To address Objective 2 (polarity analysis), we employed three bias detection tests: **WEAT** (*Word Embedding Association Test*) [5], **RND** (*Relative Norm Distance*) [16], and the **Adjectivation Test** [5]. Each technique is defined and described in detail in Section 3.4.

3.1 Target Dialect: Pajubá

The marginalized community dialect selected for this investigation was Pajubá, also known as Bajubá. It is a crypto-dialect historically developed and used by Brazilian LGBTQIAP+ communities, particularly by trans women in peripheral urban contexts [1]. Its origins are tied to the need for safe intra-community communication [2]. The dialect has a cryptic character: everyday Portuguese terms are reappropriated with meanings known only within the community. Examples include *barbie* (to refer a specific body type of gay men), *Sida* (rearrangement of the AIDS acronym used to disguise the term. Even though *SIDA* is the Portuguese acronym for AIDS, in this dialect the term is repurposed as a feminine proper name, commonly used as *Aunt Sida*), and *Katia* (to refer to alcoholic beverages). This cryptic nature was not merely a linguistic curiosity, but a survival strategy: during the Brazilian military dictatorship (1964–1985), LGBTQIAP+ individuals faced systematic political persecution, driving the community to develop new vocabulary, particularly identity markers and terms referring to its own members, as a means of communicating safely in hostile environments. Consequently, Pajubá exhibits a significant concentration of terms with sexual connotations and gender and sexuality identity markers. These properties make Pajubá a particularly compelling case for stress-testing word embeddings: its reappropriated semantics, underrepresentation in standard corpora, and socially charged vocabulary collectively expose limitations that mainstream NLP benchmarks, typically anchored in standard language varieties, are unlikely to reveal.

3.2 Dataset

The dataset used in this study was drawn entirely from the *Dicionário Aurélia* [26], a Pajubá dictionary compiling expressions from the dialect. Entries were extracted from the digital edition of the dictionary using regular-expression-based text processing. A filtering step was

applied to retain only single-word terms, excluding multi-word expressions. This decision was made to simplify the analyses and to focus on individual lexical items. This process yielded a final corpus of 564 items, comprising a mixture of dialect-native terms, standard Portuguese words with altered meanings or polarities, and vocabulary derived from African languages. For the semantic polarity analyses (Objective 2), a manual categorization was performed to identify neutral terms, those designating people, entities, and objects within the sociocultural context of the speakers. Terms were selected based on three criteria: (i) functioning as a denotative referent within the dialect (i.e., terms used to refer to entities and objects in the speakers' context); (ii) intra-group semantic neutrality; and (iii) absence of intrinsic affective polarity (i.e., terms that do not convey evaluative judgment about the referent). For instance, the word *viado*, which outside the dialect is commonly used as a pejorative term, operates within Pajubá as a neutral noun or vocative among members of the gay community, and was therefore included in this selection, whereas terms with explicit affective charge were excluded, one example is *carimbada*, a derogatory term used to refer to people living with HIV. This subset of neutral terms comprises 159 words.

3.3 Embedding Models

We employed four static word embedding models from the NILC Word Embeddings Repository [14]: Word2Vec [19], GloVe [22], Wang2Vec [17], and FastText [3]. The NILC embeddings were trained on seventeen corpora of Brazilian and European Portuguese, totalling approximately 1.4 billion tokens from diverse sources including web text, news articles, literature, and technical content, with an emphasis on formal and standard registers, a characteristic particularly relevant to the underrepresentation of Pajubá discussed in subsequent sections. All models were used in their 100-dimension configuration; replication across other available dimensionalities constitutes a natural extension of this work.

3.4 Evaluation Methodology

Biases in embedding representations can manifest in various ways. Accordingly, we conduct several complementary tests, described below.

3.4.1 Vocabulary Coverage Test

The vocabulary coverage test was implemented to assess the ability of the models to represent Pajubá dialect terms, in line with Objective (i). This analysis consisted of verifying the existence of vectors associated with the 564 unique dialect terms in each embedding model. For each term in the dataset, a lookup operation was performed in the corresponding vector space, recording the presence or absence of a representation. The overall coverage rate was computed as the ratio of terms with a vector representation to the total number of terms in the dialect vocabulary.

3.4.2 Relative Norm Distance (RND)

The *Relative Norm Distance* (RND) metric [10] quantifies semantic bias by measuring the relative association between a set of target terms and two attribute groups representing opposite semantic poles. In this study, the attribute groups were defined as: (i) a set of 25 positive terms (e.g., *respeito*, *dignidade*, *bondade*) constituting the positive semantic pole, and (ii) a set of 25 negative terms (e.g., *maldade*, *ódio*, *crueldade*) constituting the negative semantic pole. The target set comprises Pajubá identity terms (e.g., *travesti*, *bicha*, *transformista*), extracted from the subset of 159 neutral terms described in Section 3.1, whose semantic polarity we wish to evaluate.

The RND computation begins by determining the centroid vector for each attribute group. The positive centroid ($\bar{v}_{T_1}$) is computed as the arithmetic mean of the vectors of all its terms; the negative centroid ($\bar{v}_{T_2}$) is obtained analogously. The vector difference between these centroids defines the bias axis onto which the target words are projected.

All vectors, both target terms and centroids, are subsequently ℓ_2 -normalized to unit Euclidean norm, ensuring metric comparability and controlling for magnitude variation. For each target term a , its normalized projection onto the bias axis is then computed. The final RND score for the full attribute set corresponds to the arithmetic mean of these individual values.

Interpretation follows a straightforward principle: positive RND values indicate predominant association with the positive pole, while negative values indicate association with the negative pole. The absolute magnitude quantifies the strength of this association, with values near zero suggesting semantic neutrality.

In this study, the test was applied twice per model using two distinct attribute sets, both drawn from the 159-term neutral subset. The first set comprised 48 terms designating people and entities within the LGBTQIAP+ community; the second comprised 40 terms referring to objects and things. This distinction allows us to assess whether bias is specific to identity-bearing terms or extends more broadly across the dialect’s vocabulary.

3.4.3 Adjectivation Test: Qualitative Analysis of Semantic Associations

The adjectivation test is a qualitative analysis that investigates the semantic associations between identity terms and adjectives in the vector space. For this test, five central terms of the LGBTQIAP+ community were selected: *travesti*, *transsexual*, *gay*, *lésbica*, and *bissexual*.

The methodology involved three main steps. First, the **SentiLex-PT02** lexicon [6] was used as a reference resource. SentiLex-PT02 is a comprehensive inventory of Portuguese words annotated morphosyntactically and semantically, containing over 14,000 lemmas labelled with their grammatical class and affective polarity. All adjectives present in both SentiLex and the vocabularies of the embedding models under analysis were extracted, forming the candidate set for the association analysis.

Subsequently, for each target identity term, the cosine similarity between its embedding vector and all adjectives in the pre-filtered candidate set was computed, and the five closest adjectives were selected.

The objective of this test is to provide a characterisation of the semantic neighbourhood surrounding each identity term in the embedding space.

3.4.4 Word Embedding Association Test (WEAT) and SC-WEAT

The *Word Embedding Association Test* (WEAT), proposed by [5], measures semantic bias in word embeddings by testing whether two sets of target words (X and Y) are differentially associated with two sets of attribute words (A and B). Grounded in the *Implicit Association Test* (IAT) [12], WEAT treats cosine similarity between word vectors as analogous to reaction time in the IAT. For any word w , the differential association with attribute groups A and B is:

$$s(w, A, B) = \frac{1}{|A|} \sum_{a \in A} \cos(\vec{w}, \vec{a}) - \frac{1}{|B|} \sum_{b \in B} \cos(\vec{w}, \vec{b})$$

The aggregated difference across target groups and the standardized effect size d are then computed as:

$$s(X, Y, A, B) = \sum_{x \in X} s(x, A, B) - \sum_{y \in Y} s(y, A, B) \quad d = \frac{\mu_X - \mu_Y}{\sigma}$$

where μ_X , μ_Y are the mean associations for each target group and σ is the pooled standard deviation. Statistical significance is assessed via a permutation test over all possible redistributions of $X \cup Y$.

However, WEAT requires X and Y to be of similar size and represent well-defined antagonistic poles. Some studies on sexual orientation bias construct these poles using terms such as *heterosexual* and *homosexual* [13], a valid approach for binary distinctions. The present work, however, investigates a broader spectrum of gender identities and sexualities for which no natural counterpart set exists, making such a bipolar structure incompatible with the problem studied.

This work therefore adopts the *Single-Category Word Embedding Association Test* (SC-WEAT), also proposed by [5] and applied to Portuguese embeddings by [7]. SC-WEAT operates on a single target set X , computing for each $w \in X$ the effect size:

$$ES(\vec{w}, A, B) = \frac{\text{mean}_{a \in A} \cos(\vec{w}, \vec{a}) - \text{mean}_{b \in B} \cos(\vec{w}, \vec{b})}{\text{std}_{x \in A \cup B} \cos(\vec{w}, \vec{x})}$$

Positive ES indicates association with pole A ; negative with pole B . Significance is assessed via a one-sample permutation test based on sign rearrangements [13].

The attribute sets were constructed from the *pleasant vs. unpleasant* word lists of [5], adapted for Portuguese. Since WEAT was designed for English, a language without grammatical gender, direct application to Portuguese risks introducing morphological imbalances that distort results [24, 7]. To mitigate this, each gender-inflected adjective was included in both masculine and feminine forms, with its vector represented by the arithmetic mean of both. Morphologically invariant adjectives were retained in their single form. Additionally, the number of masculine and feminine adjective forms was balanced across both attribute sets, ensuring that no pole carries a disproportionate morphological gender signal that could confound the bias measurement.

4 Results

For the sake of a clear analysis, the results are organized according to the respective tests described previously.

4.1 Coverage of Dialectal Terms

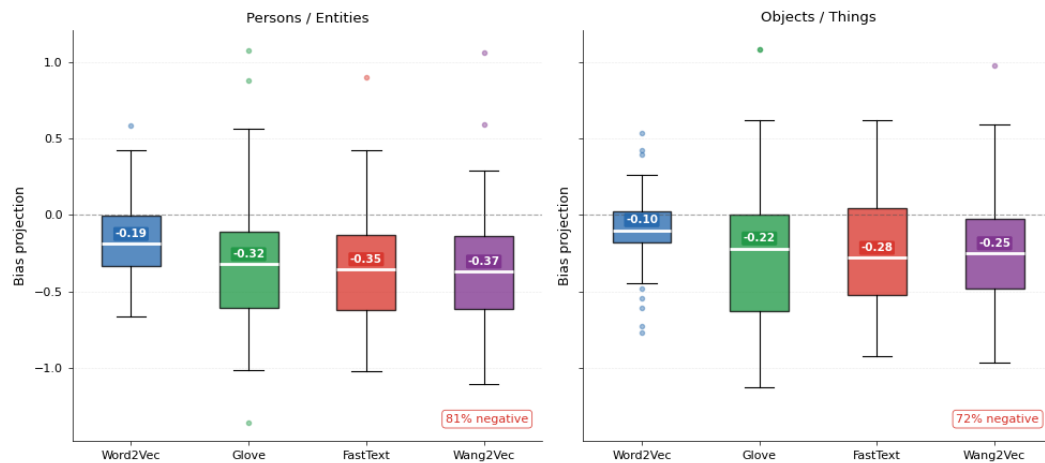
The coverage tests demonstrated that only 74% of the terms present in the reference dataset were adequately represented in the analysed embeddings. Examination of the absent terms revealed that they consist predominantly of neologisms characteristic of the dialect under study and of terms originating from African-derived dialects, with particular note given to the significant absence of high-frequency expressions from *Pajubá*, such as *aque* (money).

The terms that achieved adequate representation in the embeddings correspond primarily to vocabulary from standard Portuguese, with an emphasis on items of formal register and high frequency in standard corpora. This pattern suggests that the models capture the lexicon of written, institutionalised language far more reliably than the oral and community-specific vocabulary that characterises the dialect under investigation.

As an explanatory hypothesis for this phenomenon, the underrepresentation of dialectal markers follows directly from the low frequency of these terms in the written textual sources used to train the models. Since static word embedding models rely entirely on co-occurrence statistics derived from large written corpora lexical items that circulate primarily in oral, informal, or community-specific contexts are systematically marginalised in the resulting vector spaces.

4.2 RND (Relative Norm Distance) Test Results

Figure 1 presents the distribution of bias projections across the four embedding models for both word categories. Across all models and both categories, the median bias projection is negative, indicating a consistent tendency toward association with the negative attribute pole. This pattern is more pronounced in the People / Entities category, where 81% of projections are negative, compared to 72% in the Objects / Things category.

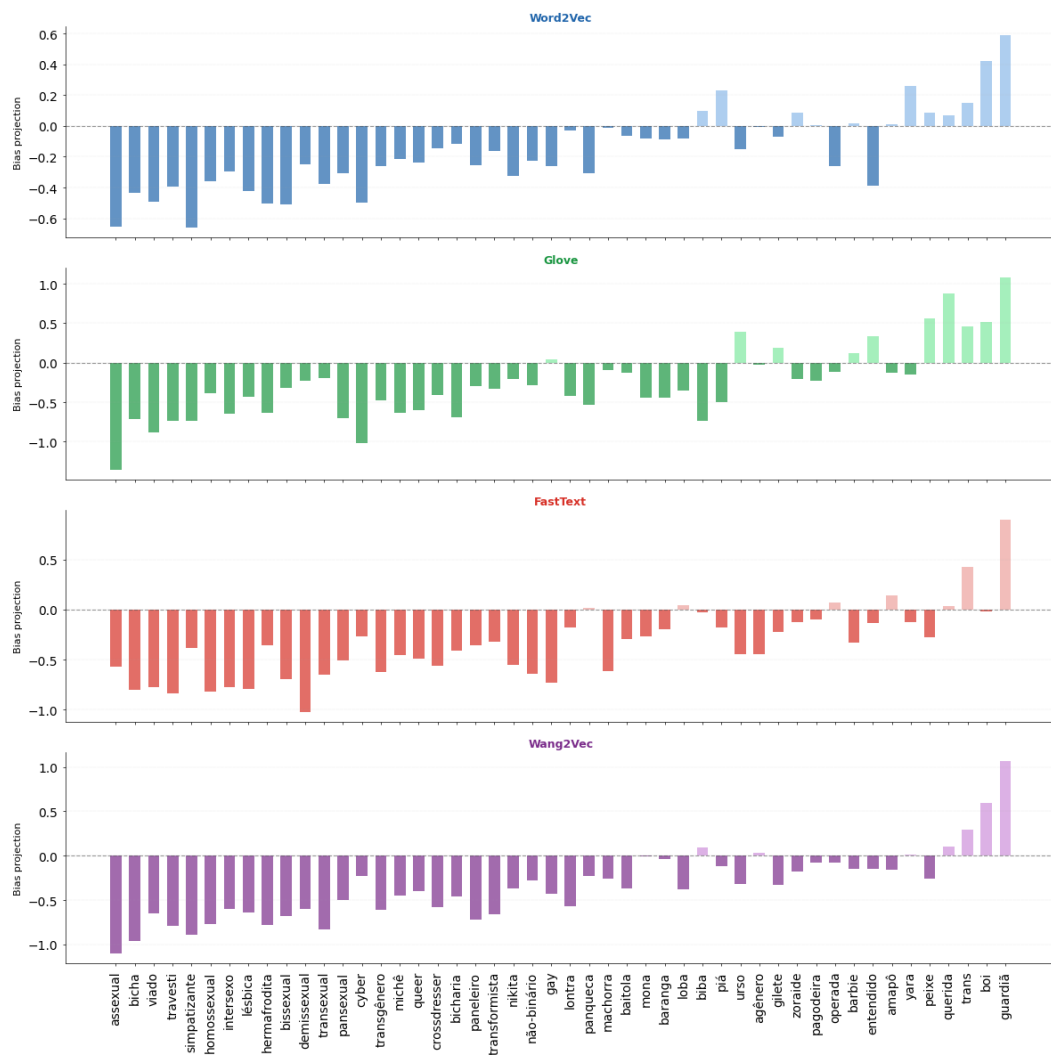


■ **Figure 1** Distribution of bias projections across embedding models.

Among the models, Word2Vec exhibits the least negative median in both categories (-0.19 for People / Entities and -0.10 for Objects / Things), alongside the widest interquartile range, suggesting greater variance in its associations. GloVe, FastText, and Wang2Vec present more concentrated distributions with stronger negative medians, ranging from -0.32 to -0.37 in the People / Entities category and from -0.22 to -0.28 in Objects / Things. Notably, FastText and Wang2Vec display the most negative central tendencies overall, with medians of -0.35 and -0.37 respectively for People / Entities.

Outliers are present across all models in both categories, with isolated positive projections reaching values above 1.0, suggesting that a small subset of terms is associated with the positive attribute pole regardless of the model. Nevertheless, these exceptions do not alter the dominant negative trend observed across the full distributions.

The consistently negative projections across models and categories suggest that the target terms, spanning gender identities and sexualities, are systematically associated with negative attributes in the embedding spaces, regardless of the specific architecture or training corpus of each model. The stronger bias observed in the People / Entities category relative to Objects / Things may reflect that terms directly denoting people are more susceptible to encoding societal stereotypes present in the training data.



■ **Figure 2** Bias projection – People / Entities.

Figure 2 presents the individual bias projections for each term in the People / Entities category across the four embedding models. The results reinforce the predominantly negative pattern observed in the aggregate analysis, with the majority of terms displaying negative projections across all models. However, a more granular examination reveals two observations.

First, while the direction of bias is largely consistent across models, the magnitude of individual projections varies considerably. Word2Vec tends to produce more moderate negative values, whereas GloVe, FastText, and Wang2Vec exhibit stronger negative projections for several terms, with some reaching values below -0.75 or even -1.0 . This suggests that although all models encode a similar directional bias, the intensity with which individual terms are associated with negative attributes is sensitive to the architecture and training corpus of each model.

Second, the subset of terms presenting positive projections is consistent across models. Terms such as, *trans*, *boi*, and most notably *guardiã* recurrently appear with positive values across all four models. This cross-model consistency suggests that these associations are not artifacts of a specific model but rather reflect patterns systematically encoded in the Portuguese language corpora used for training.

A particularly notable case within this subset is *trans*, commonly used in Portuguese to refer to transgender individuals, and which consistently presents positive projections across all models. This stands in apparent contradiction with the strongly negative projections observed for related terms such as *transsexual* and *transgênero*. However, this discrepancy is likely not a reflection of a more positive encoding of transgender identity per se, but rather a consequence of the morphological ambiguity of the term: *trans* functions as a productive prefix in Portuguese across a wide range of domains and its vector representation in the embedding space is therefore shaped by this broader and predominantly neutral or positive co-occurrence context, rather than by its community-specific usage alone.

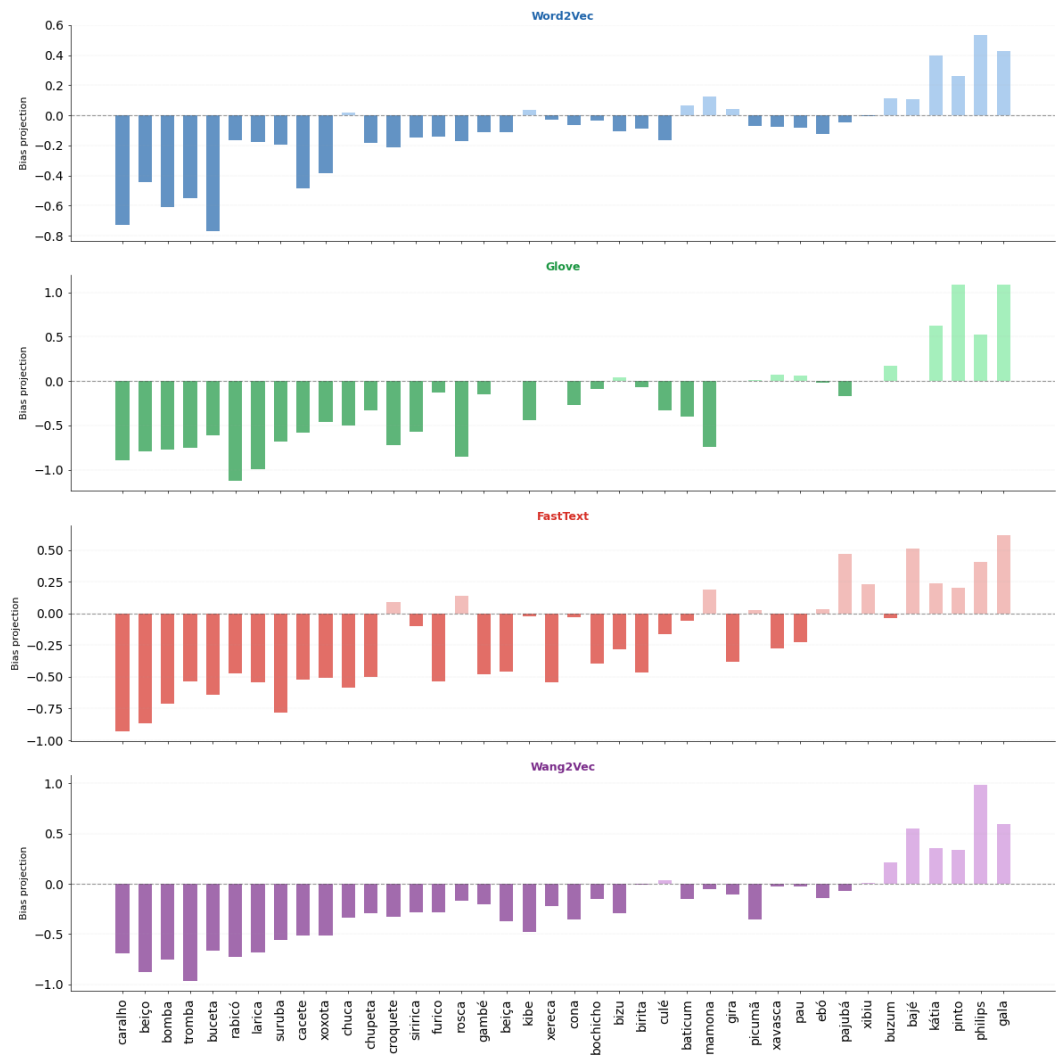


Figure 3 Bias projection – Objects / Things.

The Objects / Things category, presented in Figure 3, follows the same overall pattern, corroborating the 72% negative projection rate reported previously. The most strongly negative terms are concentrated at the left of the distribution and correspond to vocabulary with explicit sexual or anatomical connotations in Portuguese – such as *caralho*, *beico*, *bomba*, and *tromba* – whose pejorative or taboo framing is robustly captured across all models.

The positive subset comprises terms such as *philips*, *gala*, *pinto*, *kátia*, and *bajé*, several of which carry neutral or positive associations in standard Portuguese. As with *guardiã* in the previous category, their positive projections reflect formal Portuguese valence rather than community-internal reappropriation. The LGBTQIAP+ dialect operates across both ends of this spectrum, appropriating terms of positive valence while reclaiming terms of negative valence, but the models capture only the latter as negative, reflecting the dominant framing of these terms in the training data rather than their in-group meaning.

4.3 SC-WEAT Results

The SC-WEAT analysis was conducted in two stages. In the first stage, the same target terms from the RND analysis were used. It is important to note that although both methods investigate bias through the differential association of target terms with opposing attribute poles, they differ in how those poles are constructed: the RND approach uses general Portuguese positive and negative terms, while SC-WEAT contrasts a *pleasant* pole against an *unpleasant* pole based on the attribute word lists from [5], adapted for Portuguese as described in the methodology. The results of this first stage are presented in Table 1.

■ **Table 1** SC-WEAT results for the RND target set.

Model	d	p	Direction
Word2Vec	+0.5202	0.5111	Pleasant
GloVe	−0.1789	0.7580	Unpleasant
FastText	+1.0067	0.2376	Pleasant
Wang2Vec	+0.4728	0.5121	Pleasant

None of the four models yielded statistically significant results, and three out of four presented a positive effect size, suggesting a tendency toward the pleasant pole. This outcome is unexpected given the predominantly negative bias observed in the RND analysis. However, this discrepancy is likely a consequence of the composition of the target set itself: as discussed in the previous section, the RND vocabulary includes a consistent subset of terms with positive valence in standard Portuguese, such as *guardiã*, *entendido*, and *trans*, whose presence in the target set may have attenuated or reversed the aggregate effect size, masking the negative associations observed at the individual term level.

To investigate whether this tendency toward the pleasant pole was driven by the standard Portuguese terms used to refer to the community, suggesting that these identitarian markers themselves carry positive associations, or by the reappropriated vocabulary of the dialect, a second stage was conducted using a refined target set restricted to explicit identitarian terms in standard Portuguese. If the identitarian markers consistently presented negative associations while the broader dialect vocabulary presented positive ones, this would constitute evidence that the cryptic function of the dialect is being fulfilled: the reappropriated terms successfully encode positive valence in the embedding space, while the bias against LGBTQIAP+ groups persists in the standard register. The target terms were divided into two subsets: gender-related terms (*travesti*, *transsexual*, *trans*, *transgênero*, *não-binário*, *agênero*, *intersexo*, *queer*) and sexual orientation-related terms (*gay*, *lésbica*, *bissexual*, *homossexual*, *pansexual*, *asexual*, *demissexual*). The results are presented in Table 2.

The refined analysis reveals a more consistent pattern. Three out of four models present negative effect sizes for both subsets, indicating a tendency toward the unpleasant pole for terms explicitly associated with gender identity and sexual orientation. Wang2Vec

■ **Table 2** SC-WEAT results for the refined target sets by category.

Model	Gender		Orientation	
	<i>d</i>	<i>p</i>	<i>d</i>	<i>p</i>
Word2Vec	−0.4955	0.6367	−0.3565	0.8129
GloVe	−0.9512	0.3028	−0.1817	0.8681
FastText	+0.1951	0.8638	+0.7346	0.7436
Wang2Vec	−1.3163	0.2001	−0.3883	0.8439

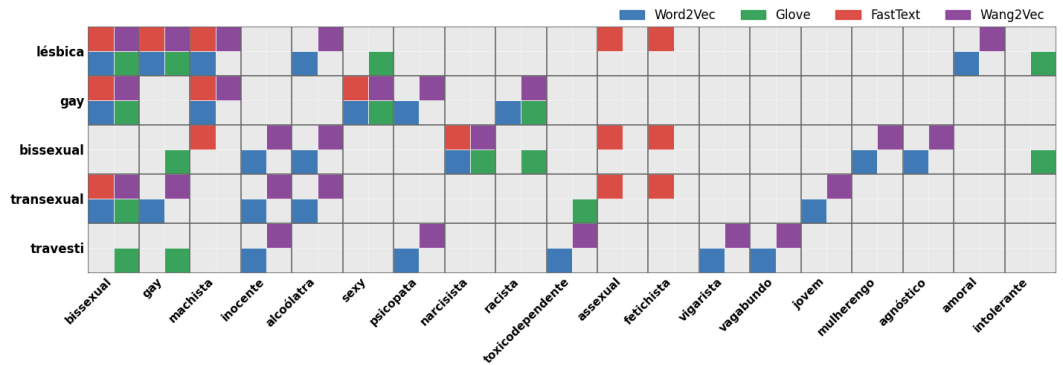
produces the strongest negative effect for the gender subset ($d = -1.3163$), followed by GloVe ($d = -0.9512$), while Word2Vec and FastText present more moderate values. FastText constitutes the sole exception, yielding positive effect sizes for both subsets, consistent with its behaviour in the first stage.

None of the results reach statistical significance, which warrants careful interpretation. The effect sizes observed, particularly for Wang2Vec and GloVe in the gender subset, are nonetheless substantial in magnitude and directionally consistent with the RND findings.

The persistent exception of FastText across both stages merits attention. Unlike the other three models, FastText represents words using subword information, decomposing terms into character n -grams before computing embeddings. This architecture makes it more robust to out-of-vocabulary terms but also means that the vector representation of a word is partially shaped by its morphological components. For terms such as *transexual*, *transgênero*, or *homossexual*, the subword structure may introduce associations with morphologically related but semantically distinct terms, potentially diluting or reversing the bias signal captured by the other models.

4.4 Adjectivation Test Results

The results of the lexical association test revealed consistent patterns of stigmatisation and identity reductionism across all models and terms analysed. For each of the five identity terms investigated (*travesti*, *transexual*, *bissexual*, *gay*, *lésbica*), two main phenomena were observed across all models: the recurrent association with other identity markers, and the attribution of pejorative terms. Figure 4 illustrates the word associations observed for each model.



■ **Figure 4** Adjectivation test results across all four embedding models.

The first phenomenon manifested through the conflation of distinct identities: *bissexual* was associated with *gay* across all models, and *lésbica* with *bissexual* in the majority of cases. This pattern suggests that the models fail to represent these identities as semantically independent, instead collapsing them into a shared neighbourhood of the embedding space.

The second phenomenon was evidenced by the systematic presence of negative attributes, including terms associated with criminality (*vigarista*, *bandido*, *delinquente*, *vagabundo*), pathologisation (*psicopata*, *alcoólatra*, *toxicodependente*, *maníaco-depressivo*), and negative moral judgement (*infiel*, *preconceituoso*, *amoral*, *intolerante*).

Certain term-specific patterns merit particular attention. The term *travesti* was associated with criminality-related expressions in three of the four models analysed. A plausible explanation for this pattern lies in the nature of the training data and the prevalence of media coverage of hate crimes against the trans population in the corpora. This discussion is elaborated in the following section.

5 Conclusion

The results obtained across the three analytical approaches consistently point to the presence of negative associations for vocabulary related to LGBTQIAP+ gender identities and sexual orientations in static Portuguese word embeddings, a pattern that persists across four architecturally distinct models and three complementary methods.

With respect to Objective 1, the vocabulary coverage test revealed that only approximately 74% of the dialect terms present in the primary source were found in the embedding models' vocabularies. This result indicates that static Portuguese word embeddings have limited representational capacity for the LGBTQIAP+ dialect, and that a substantial portion of this vocabulary remains inaccessible to embedding-based analysis. The terms that do appear are, by definition, those that have managed to cross the social boundaries of the speaker community and enter broader Portuguese language use. The terms that remain absent are likely those whose circulation is most strictly confined to in-group contexts, introducing a blind spot into any investigation that relies on these models.

With respect to Objective 2, the polarity analysis consistently pointed toward negative associations for the target vocabulary across all three methods and all four models. The models fail to capture the internal valence system of the LGBTQIAP+ dialect: terms that carry neutral polarity or that were reclaimed meaning within the community are not represented as such in the embedding space. Instead, the models reflect the dominant register of the training corpora, which frames these identities and their associated vocabulary through a predominantly negative lens. This is not a failure of the models per se, but an inherent consequence of training on data that disproportionately represents hegemonic discourse.

Other findings was that during the SC-WEAT, the aggregate tendency toward the pleasant pole appeared to contradict the RND findings. However, when the analysis was restricted to explicit identitarian terms in standard Portuguese, the negative tendency was restored across the majority of models. This divergence is interpretable as indirect evidence that the cryptic function of the dialect is partially reflected in the embedding space: reappropriated terms carry their standard Portuguese valence into the models, while the vocabulary used to explicitly name LGBTQIAP+ identities remains negatively encoded. It must be acknowledged, however, that the precise boundaries between positive and negative poles remain conceptually complex. The attribute word lists used to construct the SC-WEAT poles carry their own cultural assumptions, and the notion of what constitutes a pleasant or unpleasant association is not universally fixed. Refining these conceptual boundaries will be an important methodological challenge for future work in this area.

The adjectivation test adds a qualitative dimension to this picture. The recurrent co-occurrence of identity terms with vocabulary related to criminality, pathologisation, and moral condemnation is interpretable as evidence that these identities constitute a semantic pole in the embedding space, one that overlaps with these negative domains. A plausible hypothesis to account for this overlap lies in the social reality of LGBTQIAP+ communities in Brazil: the country consistently records the highest absolute number of lethal hate crimes against LGBT individuals in the world [20], and trans women are disproportionately relegated to sex work, a reality that is itself reflected in the dialect and in the contexts where this vocabulary circulates. Given that the training corpora include a substantial proportion of journalistic texts, the repeated reporting of violence, crime, and marginalisation involving these identities may have shaped their vector representations in the embedding space. It is therefore important to note that this does not imply that the models encode these identities as inherently criminal or pathological, but rather that these identities appear, in the training data, in contexts where such terms are present. The distinction between associative context and attributed characteristic is significant, and constitutes one of the most challenging interpretive dimensions of bias measurement in word embeddings.

5.1 Limitations and Future Work

This study is subject to several limitations that point toward directions for future work. The absence of approximately 24% of the dialect terms from the models' vocabularies represents a fundamental constraint, rooted in the scarcity of digitally accessible textual production by and about marginalised communities. Expanding the data sources through community-engaged collection or collaboration with LGBTQIAP+ organisations remains an important challenge. The small size of the SC-WEAT target sets further constrains statistical power, an inherent tension in analyses of minority linguistic communities.

The use of static embeddings, which assign a single vector to each word regardless of context, makes it impossible to distinguish between pejorative and reclaimed usages of the same term. Extending this analysis to contextual models such as BERTimbau [25] would allow for a more nuanced investigation of bias at the contextual level. The behaviour of FastText in the SC-WEAT analysis, consistently diverging from the negative tendency observed in the other models, may also suggest a bias dilution effect attributable to its subword architecture, a hypothesis that warrants dedicated investigation.

Finally, bias measurement in language models intersects with subjective, culturally situated questions of identity and valence that no single study can exhaust. As language evolves and the textual presence of LGBTQIAP+ communities grows, the biases encoded in future models may differ substantially from those observed here, making longitudinal follow-up a natural next step.

This study is intended as a first step toward a broader research agenda on bias in Portuguese language models for minority and marginalised linguistic communities, and we hope it encourages further investigation into the sociolinguistic dimensions of NLP fairness beyond the standard language register.

References

- 1 Karylleila dos Santos Andrade, Sheila de Carvalho P. Gonçalves, Filipe Porto, and Luciana C. e Silva Andrade. *Bajubá: linguagem de grupo lgbt como representação sócio-histórica e cultural*. *Desafios – Revista Interdisciplinar da UFTO*, 5(4):36–46, 2018.
- 2 Gabriela Costa Araujo. *Bajubá: memórias e diálogos das travestis*. Paco Editorial, Jundiaí, SP, 1 edition, 2019.

- 3 Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. doi:10.1162/tac1_a_00051.
- 4 Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, volume 29, pages 4349–4357, 2016. URL: <https://proceedings.neurips.cc/paper/2016/hash/a486cd07e4ac3d270571622f4f316ec5-Abstract.html>.
- 5 Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, April 2017. doi:10.1126/science.aal4230.
- 6 Paula Carvalho, Mário J. Silva, Carlos Costa, and Luís Sarmiento. Sentilex-pt: Principais características e potencialidades. *Oslo Studies in Language*, 7(1):e1444, 2015. doi:10.5617/osla.1444.
- 7 Fernanda Tiemi de Souza Taso, Valéria Quadros dos Reis, and Fábio Viduani Martinez. Analyzing discourses in Portuguese word embeddings: A case of gender bias outside the English-speaking world. *Journal on Interactive Systems*, 16(1), 2025. doi:10.5753/jis.2025.5958.
- 8 Fatma Elsafoury, Steven R. Wilson, Stamos Katsigiannis, and Naeem Ramzan. SOS: Systematic offensive stereotyping bias in word embeddings. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1263–1274, Gyeongju, Republic of Korea, 2022. International Committee on Computational Linguistics. URL: <https://aclanthology.org/2022.coling-1.108>.
- 9 Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. Understanding undesirable word embedding associations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1696–1705, Florence, Italy, 2019. Association for Computational Linguistics. doi:10.18653/v1/P19-1166.
- 10 Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644, 2018. doi:10.1073/pnas.1720347115.
- 11 Hila Gonen and Yoav Goldberg. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 609–614, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi:10.18653/v1/N19-1061.
- 12 Anthony G. Greenwald, Debbie E. McGhee, and Jordan L. K. Schwartz. Measuring individual differences in implicit cognition: The implicit association test. *Journal of Personality and Social Psychology*, 74(6):1464–1480, 1998. doi:10.1037/0022-3514.74.6.1464.
- 13 Bruce Pengfei Han. PsychWordVec: Word embedding research framework for psychological science. R package, 2025. URL: <https://psychbruce.github.io/PsychWordVec/>.
- 14 Nathan Hartmann, Erick Fonseca, Christopher Shulby, Marcos Treviso, Jéssica Silva, and Sandra Aluísio. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In *Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology*, pages 122–131, Uberlândia, Brazil, October 2017. Sociedade Brasileira de Computação. URL: <https://aclanthology.org/W17-6615/>.
- 15 Svetlana Kiritchenko and Saif M. Mohammad. Examining gender and race bias in two hundred sentiment analysis systems. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 43–53, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi:10.18653/v1/S18-2005.
- 16 Neeraja Kirtane and Tanvi Anand. Mitigating gender stereotypes in Hindi and Marathi. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 145–150, Seattle, Washington, July 2022. Association for Computational Linguistics. doi:10.18653/v1/2022.gebnlp-1.16.

- 17 Wang Ling, Chris Dyer, Alan W. Black, and Isabel Trancoso. Two/Too simple adaptations of Word2Vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi:10.3115/v1/N15-1142.
- 18 Vijit Malik, Sunipa Dev, Akihiro Nishi, Nanyun Peng, and Kai-Wei Chang. Socially aware bias measurements for Hindi language representations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1041–1052, Seattle, United States, July 2022. Association for Computational Linguistics. doi:10.18653/v1/2022.naacl-main.76.
- 19 Tomáš Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013. NIPS 2013. URL: <https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality>.
- 20 Luiz Mott, Eduardo Michels, and Paulinho. Homicídios da população de lésbicas, gays, bissexuais, travestis, transexuais ou transgêneros (LGBT) no Brasil: uma análise espacial. *Ciência & Saúde Coletiva*, 25(6):2195–2208, 2020. doi:10.1590/1413-81232020256.19672019.
- 21 Orestis Papakyriakopoulos, Simon Hegelich, Juan Carlos Medina Serrano, and Fabienne Marco. Bias in word embeddings. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 446–457, 2020. doi:10.1145/3351095.3372843.
- 22 Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi:10.3115/v1/D14-1162.
- 23 Alexandre Puttick, Catherine Ikae, Carlotta Rigotti, Eduard Fosch-Villaronga, Mark W. Kharas, Roger A. Søraa, and Mascha Kurpicz-Briki. A systematic review of bias detection methods for non-English word embeddings and language models. *Artificial Intelligence Review*, 58:370, 2025. doi:10.1007/s10462-025-11375-8.
- 24 Saghar Omrani Sabbaghi and Aylin Caliskan. Measuring gender bias in word embeddings of gendered languages requires disentangling grammatical gender signals. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 518–531, 2022. doi:10.1145/3514094.3534176.
- 25 Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. BERTimbau: Pretrained BERT models for Brazilian Portuguese. In *Intelligent Systems. BRACIS 2020*, volume 12319 of *Lecture Notes in Computer Science*, pages 403–417. Springer, 2020. doi:10.1007/978-3-030-61377-8_28.
- 26 Angelo Vip and Fred Libi. *Aurélia, a Dicionária da Língua Afada*. Editora do Bispo, São Paulo, Brazil, 2006.

Gender Bias Evaluation in English-Portuguese Automated Translation Outputs

Xiaolan Xu 

Faculty of Arts and Humanities, University of Lisbon, Portugal

Sara Mendes 

Center of Linguistics of the University of Lisbon and Faculty of Arts and Humanities,
University of Lisbon, Portugal

Abstract

Automated translation (AT) plays a pivotal role in breaking down language barriers and fostering cross-cultural communication. However, gender bias in AT remains a pressing concern, particularly for language pairs where the source language generally does not have grammatical gender and the target language does. In this context, the present research aims to assess gender bias in English-Portuguese AT outputs. Two machine translation (MT) systems and two LLMs are evaluated in this study, examining the correlation between translated gender and societal stereotypes, and how the different models handle nouns whose gender is unknown. The outputs of two commercial MT systems (Google Translate and DeepL Translator) and two general-purpose LLMs (ChatGPT-4o and DeepSeek-V3), translating from English into European Portuguese, were analyzed. We used a subset of the WinoMT challenge set to assess gender rendering accuracy across three conditions: gender-defined nouns, gender-undefined nouns, and anaphoric pronoun resolution. Our main goal is therefore to evaluate the accuracy of gender information transmission and to understand the extent to which gender bias is present in AT outputs. Our results indicate that all four models struggle to faithfully convey gender information from source to target, with male-centric outputs predominating across conditions and MT systems showing a more pronounced bias. This work contributes a preliminary cross-system comparison for an underexplored language pair and lays the groundwork for larger-scale evaluations of gender-inclusive AT.

2012 ACM Subject Classification Computing methodologies → Machine translation; Social and professional topics → Gender

Keywords and phrases Gender Bias, Large Language Models (LLMs), Neural Machine Translation (NMT), English-Portuguese Translation

Digital Object Identifier 10.4230/OASICS.SLATE.2026.8

Supplementary Material *Dataset:* <https://github.com/xu-xiaolan/gender-bias-en-pt-at> [20]
archived at `swb:1:dir:f3d848b15460545b49033fd09dfa2af692d74510`

Funding We acknowledge the support of Fundação para a Ciência e a Tecnologia (UID/00214: Centro de Linguística da Universidade de Lisboa).

1 Introduction

In our globally interconnected world, machine translation (MT) plays a pivotal role in breaking down language barriers and fostering cross-cultural communication. After decades of development, since the concept of MT was created, the current state-of-the-art MT system is neural machine translation (NMT). NMT uses deep neural networks to model the translation process end-to-end. Unlike earlier models like rule-based machine translation (RBMT) or statistical machine translation (SMT), which relied on separate components for tasks such as language modeling and phrase alignment, NMT learns to translate directly from source to target language using large datasets. This results in more fluent and contextually accurate translations. Nowadays, with the development of technology, Large Language Models (LLMs) perform very well in text processing, including translating text.



© Xiaolan Xu and Sara Mendes;

licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 8; pp. 8:1–8:14

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

As highly data-driven approaches, NMT and LLMs rely heavily on extensive datasets and corpora for effective training and learning. However, this reliance on data introduces a significant challenge: the inherent biases present in natural language data. For instance, gender bias often permeates training datasets, with a prevalence of male-centric examples over female-centric ones. Consequently, systems trained on such data may inadvertently perpetuate or even amplify these biases [21].

Gender bias can be manifested in various ways, from the use of gendered pronouns to the translation of gender-specific terms, resulting in errors and reducing output quality, especially when the target language has grammatical gender, like Portuguese. In languages that have grammatical gender distinctions, automated translation (AT) systems may assign masculine pronouns by default or prioritize male-centric interpretations of ambiguous expressions, e.g., translating secretaries and nurses as females, and programmers and doctors as males, regardless of context. This distorts the intended meaning of the original text, leading to incorrect translations. Therefore, addressing gender bias in AT systems is not just a matter of linguistic accuracy but also a crucial step toward improving the performance of automated translation.

Gender bias in machine translation has attracted a lot of attention. Stanovsky et al. [18] present the first challenge set and evaluation protocol to measure the effect of gender bias in MT, and this research reveals the obvious existence of gender-biased translation inaccuracies in four widely used industrial MT systems and two contemporary academic MT. Other researchers have outlined different strategies to reduce gender bias in MT, such as using word embeddings [6] and transfer learning [15], or creating a new learning framework [5]. These studies involve many languages, such as English, German, Spanish, Italian, Finnish, Turkish, Korean, etc., and the translation language pairs mainly involve English. Meanwhile Portuguese has not attracted much attention, while gender expressions in English and Portuguese are highly asymmetrical: in English, there is only lexical gender (*mother/father*), and pronominal gender (*she/he/it*); while in Portuguese, nouns have morphological endings to express the gender class as masculine or feminine (*amigo/amiga*), and due to morphosyntactic agreement, several parts of speech besides the noun carry gender inflections (e.g., determiners, adjectives).

In this context, the present research aims to assess gender bias in English-Portuguese AT outputs. It assesses gender bias in the outputs produced by two commercial NMT systems (i.e., Google Translate and DeepL) and two general-purpose LLMs (i.e., ChatGPT-4o and DeepSeek-V3), with the objectives of evaluating the accuracy of gender information transmission in English-Portuguese AT, analyzing the correlation between gender translation and societal stereotypes, and examining how gender-neutral or unknown gender phrases are handled in AT.

This research is designed to answer three research questions:

- (i) Do selected systems correctly convey the gender information of the source text in English-Portuguese AT?
- (ii) Does the translation output reproduce or amplify gender stereotypes?
- (iii) How do the systems tested handle the unknown or neutral gender from the source to the target?

These research questions will allow us to assess how contemporary automated translation systems handle gender representation. The remainder of this work is structured as follows: Section 2 reviews prior work on gender bias in MT and outlines the gender relevant grammatical features of English and Portuguese. Section 3 details our methodological approach, including data selection and annotation aspects. Section 4 presents the data analysis and findings, and section 5 offers the conclusion.

2 Background

The theoretical background for this study necessarily includes existing research on gender bias in automated translation (Section 2.1), as well as the linguistic description of grammatical gender systems of English and Portuguese (Section 2.2).

2.1 Gender bias in AT

Gender bias is a significant issue in AT, leading to ongoing research efforts in developing bias mitigation techniques. This phenomenon may lead to misgendered translations, resulting, among other harms, in the perpetuation of stereotypes and prejudices.

Gender biases in AT can be manifested by, for example, producing disproportionately fewer feminine forms or consistently translating professional titles in a gender-stereotypical manner [1]. These biases lead to inaccuracies in AT outputs, requiring the post-editing of AT outputs to correctly reflect feminine gender. This results in greater time and effort, which in turn leads to higher financial costs [16].

Stanovsky et al. [18] present the first challenge set, called WinoMT, and the corresponding evaluation protocol to measure the effect of gender bias in MT. This research reveals that four widely used industrial MT systems and two contemporary state-of-the-art academic MT models exhibit notable gender-biased translation inaccuracies across all the examined target languages [18].

Although a systematic study of such biases can be difficult, Prates et al. [11] believe that MT tools can be exploited through gender neutral languages to yield a window into the phenomenon of gender bias in AI. Prates et al. [11] use it to build sentences in constructions like “He/She is an Engineer” (where “Engineer” is replaced by different occupation target nouns of interest) in 12 different gender-neutral languages such as Hungarian, Chinese, Yoruba, and several others. Font et al. [6] take advantage of the fact that word embeddings are used in NMT to propose a method to equalize gender biases in neural machine translation using these representations. They experiment and analyze the integration of two debiasing techniques over GloVe embeddings in the Transformer translation architecture. Cho et al. [2] propose a scheme for making up a test set that evaluates gender bias in a MT system working with Korean, a language with gender-neutral pronouns.

Saunders and Byrne [15] use transfer learning on a small set of trusted, gender-balanced examples to reduce gender bias, and solve the “catastrophic forgetting” problem in adaptation and in inference. Fleisig et al. [5] present a learning framework that alleviate gender bias in seq2seq machine translation. Costa-jussà et al. [4] discuss from an algorithmic perspective whether the chosen architecture, when trained with the same data, influences the level of gender bias, and they propose two methods for interpreting and studying gender bias in MT based on source embeddings and attention.

Recent work has begun to examine gender bias in LLMs, revealing that base LLMs exhibit a higher degree of gender bias than NMT models in English-to-Catalan and English-to-Spanish translation [14]. Similarly, Popović and Lapshinova-Koltunski [10] show that ChatGPT introduces different gender bias patterns when compared to MT systems. These findings suggest that model architecture plays a meaningful role in bias behavior, motivating cross-system comparative analyses.

■ **Table 1** English pronouns [19, p. 13].

		Personal pronouns		Possessive pronouns		Reflexive pronouns
		subjective case	objective case	determiner function	nominal function	
1p	sg.	I	me	my	mine	myself
	pl.	we	us	our	ours	ourselves
2p	sg.	you		your	yours	yourself
	pl.					yourselves
3p	masc.	he	him	his		himself
	sg. fem.	she	her		hers	herself
	non-personal	it		its		itself
	pl.	they	them	their	theirs	themselves

2.2 Grammatical gender in English and Portuguese

Gender is defined as “classes of nouns reflected in the behavior of associated words [7, p. 231]”. In other words, when a language has, for instance, three genders, it means that the nouns in that language can be divided into three distinct classes [3, p. 4]. These classes are syntactically identifiable through the agreement patterns expressed within the nominal phrase (determiners, adjectives), outside the nominal phrase (pronouns, verbs, predicates) and partially also through the noun triggering the agreement itself [17, p. 8]. Therefore, gender functions as a morphosyntactic feature of language since it is required for agreement [8].

In English, there are three genders: masculine, feminine and neuter. These three genders are categorized based on semantics, the criteria being humanness and the sex of the relevant referents, for example, *boy*, *girl*, *dog*, *stone* and *love*. This type of gender assignment system is a semantic gender assignment system [8]. The only domain of grammar where gender is visible in English, i.e. where agreement is triggered, is the pronominal system [17]. As shown in Table 1, it is obvious that only the singular 3rd person has three different forms for different genders, in which masculine pronouns are used for male human referents (*he/him/his/himself*), feminine pronouns for human female referents (*she/her/hers/herself*) while neuter is used for all other entities (*it/its/itself*)¹. In their anaphoric use, these pronouns replace the relevant nouns.

It is also important to note that in English, there is a group of lexical items that can refer to either a male or female representative of the human species, for example, *developer*, *hairstylist*, *driver*, *housekeeper*, *farmer*, etc. These nouns are classified as “personal dual gender” [12].

Siemund [17, p. 161] argues that for these nouns, the choice of pronouns is determined by the referent rather than the noun itself, and that this group includes many profession-related terms.

Since the choice of pronouns depends on the referent rather than on the nouns themselves, pronouns in co-reference relations with such nouns can, in turn, reveal the gender of the referent associated with them. This study focuses precisely on such lexical items.

Conversely, Portuguese has a grammatical gender system with two categories: masculine and feminine. These two genders are “semantically consistent in that nouns referring clearly to males are masculine (*aluno/gato*_[masc.]), those referring clearly to females are feminine

¹ The use of pronouns discussed here is limited to general rules. In actual language use, there are exceptions. For instance, animals, though not human, may be referred to as “he” or “she” based on their biological sex. Similarly, in the case of inanimate objects like “ship”, the feminine pronoun “she” is often used. These usages are influenced by emotional and cultural factors, which will not be explored in detail here.

(*aluna/gata*_[fem.]); but the gender of other nouns is for the most part arbitrary (*copo*_[masc.], *janela*_[fem.]) [7, p. 232]”. This type of system is referred to as a semantic-and-formal gender assignment system, where the assignment is based on both meaning and form – specifically phonological and morphological features, or a combination of both [8].

Gender is an inherent property of nouns, and, as mentioned earlier, gender agreement occurs both within and beyond the nominal phrase. In English, gender agreement is limited to the pronominal system. In contrast, Portuguese exhibits a broader agreement domain, including pronouns (see (1a)), determiners (1b), adjectives (1c), numerals (1d), and verb participles (1e) [13].

- (1) (a) *A Ana viu um gato bonito e ela gostou dele.* (Ana saw a_[masc.] beautiful_[masc.] cat_[masc.] and she liked it_[masc.].)
 (b) masculine determiners: *o gato* (the cat), *os gatos* (the cats)
 feminine determiners: *a gata* (the cat), *as gatas* (the cats)
 (c) masculine adjectives: *um trabalho duro* (hard work)
 feminine adjectives: *uma tarefa dura* (hard task)
 (d) common cardinal numerals in masculine: *vinte e dois/duzentos ovos* (twenty-two/two hundred eggs)
 common cardinal numerals in feminine: *vinte e duas/duzentas bananas* (twenty-two/two hundred bananas)
 (e) verbal passive clauses: *O portão foi aberto/A janela foi aberta pelo gato.* (The gate_[masc.] was opened_[masc.]/The window_[fem.] was opened_[fem.] by the cat.)

To sum up, English and Portuguese differ significantly in how they express gender. The linguistic structural differences introduce challenges in maintaining accurate and unbiased gender representation during AT, especially when translating from a less gendered language (English) into a highly gendered one (Portuguese).

3 Methodology

This study involved three main stages: translation, annotation, and data analysis, incorporating both quantitative and qualitative methods. First, we translated 120 original sentences from the WinoMT dataset [18], using each of the four selected models, applying the same input segments to ensure consistency in the comparison (see section 3.1). Translations were performed sentence by sentence, as the WinoMT dataset consists of isolated sentences specifically designed to test local contextual interpretation – particularly of pronouns and semantic roles. In this study, we focus on the role of pronouns in the generation of accurate gendered translations in European Portuguese.

As aforementioned, the four models considered in this work include two commercial NMT systems (Google Translate² and DeepL Translator³) and two LLMs (ChatGPT-4o⁴ and DeepSeek-V3⁵). These systems were selected based on their popularity, demonstrated translation quality, technological reliability, and free public accessibility. To ensure the uniformity, both MT systems were set to output translation in European Portuguese, and both LLMs received the same prompt: “Please translate this sentence into European Portuguese.” This controlled for the variant of the language output produced and ensured comparability

² <https://translate.google.com/>

³ <https://www.deepl.com/translator>

⁴ <https://chatgpt.com/>

⁵ <https://chat.deepseek.com/>

across systems. The overall goal of this methodology is to assess how each system handles gender-sensitive structures during translation, particularly with regard to grammatical gender in European Portuguese.

The four systems serve as the independent variable, allowing for a systematic comparison of their outputs under controlled conditions. The dependent variables are the features annotated for each translation output: the grammatical gender assigned to gender-defined and gender-undefined occupation nouns, the stereotypical or anti-stereotypical nature of those gender assignments, and the recoverability of gender information encoded in source pronouns in the translation. Several conditions were kept constant across all systems to ensure comparability: all systems received the same 120 source sentences, they were set to translate to the same target language variant (European Portuguese), and, in the case of LLM-based translations, the same prompt was used with all models tested.

In the following subsection 3.1, we discuss the selection of the dataset and identify linguistic elements which carry gender information. In subsection 3.2, we describe the annotation process.

3.1 Dataset selection and gender-based structural properties

In this study, we used 120 sentences from WinoMT [18] to test four models with regard to gender bias. WinoMT is a challenge set developed to test gender bias in MT systems, containing a total of 3,888 sentences in English.

The sentences in WinoMT follow a structured pattern, in which they describe a scenario that includes two characters represented by occupation nouns (e.g., *the doctor* and *the nurse*), using a personal pronoun to indicate the gender of one of them (e.g., *he/she/him/her*), for example: *The developer argued with the designer because she did not like the design.*

These sentences are structurally ambiguous with respect to pronoun reference: the pronoun could, in principle, refer to either of the two characters mentioned previously in the sentence. However, for human readers, context and real-world knowledge typically give rise to a preferential interpretation. MT systems and LLMs face a particular challenge in this regard, as they must produce a grammatically marked translation without the pragmatic and cultural resources that guide human comprehension. Although in the sentences of this corpus it is usually only the gender of one of the characters that is indicated by means of the pronoun, due to the asymmetry in the expression of grammatical gender between English and Portuguese (as introduced in section 2.2), when translating from English to Portuguese, the model necessarily has to decide the gender of both characters.

This makes WinoMT a sensitive instrument for detecting gender bias: when a model defaults to a stereotypical gender assignment rather than following the cue provided by the context, the bias becomes directly observable in the translation output.

In this work, 120 sentences were randomly sampled from WinoMT for use as source texts in this study. This subset was chosen to conduct a preliminary evaluation of gender bias in English-to-Portuguese AT, with the aim of identifying relevant patterns and key influencing factors to inform more extensive future evaluations.

Using the occupational stereotype framework from Levy et al. [9], which has been annotated and human-validated, we determined the stereotype directionality of the gender-defined occupation nouns in the selected 120 sentences. In all these sentences, the occupation referred to via gendered pronouns is anti-stereotypical (e.g., *The mechanic gave the clerk a present because she won the lottery*, where *mechanic* is the antecedent of *she*). The 120 sentences consist of 60 pairs of sentences, yielding a balanced distribution of 60 male-targeted and 60 female-targeted instances. The full list of occupations involved is provided in Appendix A.

3.2 Annotation

In the annotation step, we annotated both the source texts and the target texts.

For the source texts, in addition to annotating the gender stereotype of profession nouns occurring in our data, we also annotated nouns whose gender is not defined in the sentence. For example, in the sentence “The mechanic gave the clerk a present because she won the lottery,” “the mechanic” is in a co-reference relation with the personal pronoun “she”, indicating a female mechanic, which is anti-stereotypical. Meanwhile, the gender of “the clerk” is not determined in the sentence, so we annotated it as an undefined gender noun in this sentence.

Regarding the target texts, we performed fine-grained manual annotation along three dimensions: gender-defined nouns, gender-undefined nouns, and pronouns.

Firstly, for gender-defined nouns, we evaluated whether the grammatical gender in the target text accurately reflected the intended gender of the noun in the source text (e.g., whether “the mechanic” was correctly translated as “*a mecânica*”_[fem.]). We annotated the gender in the translation based on the form of the target noun and morphological agreement (e.g., *a professora/a gerente* as feminine, *o médico/o assistente* as masculine). With the occupational stereotype framework from Levy et al. [9], we also annotated whether the gender of the nouns in the translation preserved the anti-stereotypical nature of the source or reinforced a stereotype.

Secondly, for gender-undefined nouns, we annotated the gender used in the translation and its associated stereotype using the same method.

Finally, we identified gender expression in Portuguese based on grammatical and syntactic cues corresponding to the pronouns in the source texts. We distinguished two primary analytical categories: gender-recoverable and non-gender-recoverable translations. It is worth noting that gender recoverability is not limited to the morphological marking of the occupation nouns themselves, but extends to cases where gender is recoverable from the broader grammatical context.

The gender-recoverable group includes cases where the noun was expressed through a pronoun with gender marking (e.g., *ele, ela, -o, -a, este, esta*) or via a null subject accompanied by constituents with gender agreement, such as gendered adjectives, allowing us to infer the gender of the noun. In example (2), there are no nouns in the causal clause, but the gender information is made apparent in the subject complement, since *zangada* is a feminine adjective and *stressado* is a masculine adjective and they agree with the gender of the null subject. We also considered gender-marked possessive constructions as gender-recoverable, when these involve contractions of prepositions and personal pronouns (i.e., *dele* and *dela*), see (3).

- (2) (a) *A advogada gritou com o cabeleireiro porque estava zangada.* (The lawyer_[fem.] yelled at the hairdresser_[masc.] because she was mad.)
 (b) *O trabalhador da construção civil marcou uma consulta com o conselheiro porque estava stressado.* (The construction worker_[masc.] scheduled an appointment with the counselor_[masc.] because he was stressed.)
- (3) (a) *O livro dele está perdido.* (His book is lost.)
 (b) *O livro dela está perdido.* (Her book is lost.)
- (4) *O cozinheiro preparou um prato para a professora porque tinha fome.* (The cook_[masc.] prepared a dish for the teacher_[fem.] because he was hungry.)

■ **Table 2** Gender-defined nouns: translation accuracy and gender bias.

	Google Translate	DeepL	ChatGPT-4o	DeepSeek-V3
Correct (Anti-stereotype)	81 (67.5%)	70 (58.3%)	66 (55%)	62 (51.7%)
Wrong (Stereotype)	39 (32.5%)	50 (41.7%)	54 (45%)	58 (48.3%)
Gender bias Δ (M% – 50%)	+17.5%	+20.8%	–3.3%	+25%
Total	120 (100%)	120 (100%)	120 (100%)	120 (100%)

■ **Table 3** Mixed-effects logistic regression predicting gender-defined nouns translation accuracy.

Predictor	β	OR	95% CI	p
<i>Ref.: male target gender, NMT</i>				
Target gender: F	–1.724	0.18	[0.08, 0.38]	<.001 ***
System type: LLM	–0.619	0.54	[0.34, 0.86]	.009 **
$N = 480$ $AIC = 569.9$ $\sigma^2_{\text{sentence}} = 2.33$				

Conversely, the non-gender-recoverable group consists of three types: (i) possessive pronouns that agree with the possessed object rather than the possessor (e.g., *o seu livro*, *a sua casa*, where “*seu/sua*” agree with the gender of “*livro/casa*”, not the referent of a target noun that is the possessor); (ii) personal pronouns in clitic forms lacking morphological gender marking, such as *-lhe* or *-se*, which can therefore have either a male or female antecedent; and (iii) null subjects without any accompanying gender-marked elements, as shown in example (4).

The annotation process was conducted by the first author, proficient in both Portuguese and English, and validated by the second author, a native speaker of Portuguese also proficient in English.

This categorization enables us to quantify and compare how each system preserves, neutralizes, or omits gender information in the target text.

4 Results

In this section, we present and discuss the results obtained in the stages described previously. The analysis is structured into three main parts, section 4.1 examines gender-defined nouns, where the gender of the referent is explicitly indicated in the source sentences. Section 4.2 explores gender-undefined nouns, analyzing cases where gender is not overtly indicated in the source sentence. Section 4.3 focuses on pronouns, investigating their use in translation, and how they contribute to the overall gender representation in discourse.

4.1 Gender-defined nouns

Table 2 shows the results of gender translation when the referent of the target noun is explicitly determined by a gendered pronoun in the source sentence. Google Translate performed best among the systems, with 67.5% of the translations correctly preserving the overtly indicated gender. DeepL followed with 58.3%, while ChatGPT-4o and DeepSeek-V3 trailed at 55% and 51.7% respectively. These results suggest a considerable drop in accuracy from Google Translate to the other systems, highlighting challenges in handling even explicitly marked gender.

■ **Table 4** Gender-undefined nouns translation results.

	Google Translate	DeepL	ChatGPT-4o	DeepSeek-V3
Stereotype	106 (88.3%)	99 (82.5%)	91 (75.8%)	87 (72.5%)
M	60 (50%)	59 (49.2%)	53 (44.2%)	46 (38.3%)
F	46 (38.3%)	40 (33.3%)	38 (31.7%)	41 (34.2%)
Anti-stereotype	14 (11.7%)	21 (17.5%)	29 (24.2%)	33 (27.5%)
M	14 (11.7%)	20 (16.7%)	22 (18.3%)	19 (15.8%)
F	0 (0%)	1 (0.8%)	7 (5.8%)	14 (11.7%)
Δ (M% – 50%)	+11.7%	+15.9%	+12.5%	+4.2%
Total	120 (100%)	120 (100%)	120 (100%)	120 (100%)

Table 2 also presents the directionality of gender translation. Despite all source sentences being anti-stereotypical by design (i.e., feminine pronouns linked to stereotypically male professions and vice versa), all four systems showed a strong preference for translating the target noun as masculine, as reflected in the gender bias Δ . DeepSeek-V3 displayed the most extreme bias ($\Delta = +25\%$), followed by DeepL ($\Delta = +20.8\%$) and Google Translate ($\Delta = +17.5\%$). ChatGPT-4o was the only system approaching gender balance ($\Delta = -3.3\%$), with a slight feminine tendency.

Because all source instances were anti-stereotypical, any correct gender translation results in an anti-stereotypical target output. Therefore, the proportion of stereotype-aligned translations directly reflects the number of gender translation errors. DeepSeek-V3 produced the highest rate of stereotype-conforming outputs (48.3%), followed by ChatGPT-4o (45%), DeepL (41.7%), and Google Translate (32.5%). These results indicate that gender mistranslations are not random, but systematically skew toward reinforcing dominant gender stereotypes, particularly in the form of default masculine rendering of professions.

To further examine the significance of these patterns, a mixed-effects logistic regression was conducted with translation accuracy as the dependent variable, source gender and system type as fixed effects, and sentence as a random effect, shown in Table 3. Results confirmed that target gender significantly predicted translation accuracy (OR = 0.18, 95% CI [0.08, 0.38], $p < .001$), with female-targeted translations being substantially less likely to be correct than male-targeted ones. System type also emerged as a significant predictor (OR = 0.54, 95% CI [0.34, 0.86], $p = .009$), with LLM systems showing lower predicted accuracy than NMT systems overall. The substantial random effect variance ($\sigma^2 = 2.33$) justified the inclusion of sentence as a random effect.

4.2 Gender-undefined nouns

Table 4 presents the translation behavior of the four models when handling gender-undefined nouns, i.e., instances where the source text provides no explicit gender cues, requiring the model to opt for a given gender based only on the noun.

Unsurprisingly, across all systems, stereotype-aligned translations dominate overwhelmingly, ranging from 72.5% (DeepSeek-V3) to 88.3% (Google Translate). All four systems also show a masculine bias, as reflected in the positive Δ values, though DeepSeek-V3 is closest to gender balance ($\Delta = +4.2\%$). Anti-stereotypical translations remain rare across all systems, and feminine anti-stereotypical outputs are almost non-existing in MT: Google

■ **Table 5** Pronoun gender recoverability in the translation.

	Google Translate	DeepL	ChatGPT-4o	DeepSeek-V3
Recoverable	70 (58.3%)	69 (57.5%)	89 (74.2%)	116 (96.7%)
of which M	27 (22.5%)	37 (30.8%)	46 (38.3%)	58 (48.3%)
of which F	43 (35.8%)	32 (26.7%)	43 (35.8%)	58 (48.3%)
Unrecoverable	50 (41.7%)	51 (42.5%)	31 (25.8%)	4 (3.3%)
Total	120 (100%)	120 (100%)	120 (100%)	120 (100%)

■ **Table 6** Pronoun gender translation accuracy by gender.

	Google Translate	DeepL	ChatGPT-4o	DeepSeek-V3
<i>Overall (out of 120)</i>				
Correct	65 (54.2%)	65 (54.2%)	86 (71.7%)	116 (96.7%)
Wrong	5 (4.2%)	4 (3.3%)	3 (2.5%)	0 (0%)
Unrecoverable	50 (41.7%)	51 (42.5%)	31 (25.8%)	4 (3.3%)
<i>Male-defined pronouns (out of 60)</i>				
Correct (M to M)	26 (43.3%)	35 (58.3%)	45 (75%)	58 (96.7%)
Wrong (M to F)	4 (6.7%)	2 (3.3%)	2 (3.3%)	0 (0%)
Unrecoverable	30 (50%)	23 (38.3%)	13 (21.7%)	2 (3.3%)
<i>Female-defined pronouns (out of 60)</i>				
Correct (F to F)	39 (65%)	30 (50%)	41 (68.3%)	58 (96.7%)
Wrong (F to M)	1 (1.7%)	2 (3.3%)	1 (1.7%)	0 (0%)
Unrecoverable	20 (33.3%)	28 (46.7%)	18 (30%)	2 (3.3%)
Total	120 (100%)	120 (100%)	120 (100%)	120 (100%)

Translate produces none, DeepL only 1 case (0.8%). These remain particularly scarce in LLM-produced translations: DeepSeek-V3 produces the highest at 14 cases (11.7%) and GPT-4o generates 7 cases (5.8%).

Overall, these results point to a systemic default toward masculine and, as to be expected, stereotype-conforming translations in the absence of explicit gender markers. While the two LLMs show marginally more balanced outputs compared to the NMT systems, the overwhelming majority of outputs across all systems reinforce stereotyped gender-occupation associations, highlighting persistent bias in undefined gender resolution.

4.3 Pronouns

In this section, we will discuss the translation outputs of the pronoun occurring in the source texts, which is responsible for whether the pronoun is rendered indicating the gender of the target noun. We divided the outputs into two main cases: gender information is recoverable in the translation, in which case we can determine whether the pronoun is masculine or feminine; and gender information is not recoverable in the translation.

Since Portuguese allows for null subjects and various pronominal forms that may or may not encode gender, whether the gender information of the source pronoun is preserved or lost in translation directly reflects whether the model has correctly resolved the referent of the pronoun, making it a meaningful dimension of evaluation alongside the analysis of the occupation noun itself.

Table 5 presents the distribution of gender-recoverable and unrecoverable pronoun translations across outputs produced by the four models. DeepSeek-V3 demonstrates the highest recoverability rate (96.7%), with an equal distribution between male and female (both 48.3%), suggesting balanced gender preservation. ChatGPT-4o follows with 74.2% recoverability and a relatively even gender balance. In contrast, Google Translate and DeepL lag behind with recoverability rates of 58.3% and 57.5% respectively. Interestingly, Google Translate recovers female referents more often (35.8%) than male (22.5%), while DeepL shows a slight male preference.

Table 6 presents translation accuracy in the translation of pronouns and further breaks it down by gender. Overall, DeepSeek-V3 vastly outperforms other models with a 96.7% rate of correct translations and no wrong gender assignments. GPT-4o follows with 71.7%, while Google Translate and DeepL each achieve just over half of accurate pronoun translations (54.2%). Wrong gender assignments are minimal across all systems, indicating that when systems do assign gender, they mostly do so correctly.

Breaking down pronoun translation accuracy by gender, all models perform better on female-defined pronouns than male-defined pronouns, with the exception of DeepSeek-V3, which maintains a consistently high performance across both genders (96.7%). Google Translate shows the largest gender gap, with only 43.3% correct male-translations of pronouns compared to 65% for female ones. Overall, these results reveal varying degrees of gender information loss across systems, with DeepSeek-V3 demonstrating the strongest capability to preserve explicit gender cues in translation.

5 Conclusion

In this study, we used a subset of the WinoMT dataset, a challenge set for grammatical gender translations, to test and compare two machine translation systems (Google Translate and DeepL Translator) and two Large Language Models (ChatGPT and DeepSeek), for the English-European Portuguese language pair. We analyzed gender rendering in different conditions, to assess how faithfully gender information is conveyed and whether stereotypical patterns emerge or intensify.

The first research question focused on whether the models tested correctly convey the gender information from the source to the target. Given the contrasting linguistic properties of English and European Portuguese, gender information in the source is mainly conveyed by the pronoun anaphorically linked to a gender unmarked noun in the sentence. The discrepancies observed indicate that gender rendering is not uniformly handled across different part-of-speech, and while some systems may perform better with explicit lexical cues, they may struggle with pronoun resolution or vice versa.

Secondly, our results show that translation outputs do indeed reproduce and, in some cases, amplify gender stereotypes. In the translations of gender-defined nouns (Table 2), most models lean toward producing male target nouns in Portuguese, particularly DeepSeek-V3 (75% male) and DeepL (70.8% male), whereas GPT-4o was the only system to produce a more balanced distribution. This male-dominant translation is more evident when considered alongside the stereotype classification in Table 2: all four models produced translations reproducing gender stereotypes in at least one-third of the cases, with DeepSeek-V3 reaching the highest proportion (48.3%).

The trend continues when analyzing gender-undefined nouns (Table 4). In this scenario, all systems overwhelmingly defaulted to gender-stereotypical translations, especially masculine-stereotype. Unsurprisingly, this suggests that without explicit gender cues, the four models tend to rely heavily on stereotypical associations between occupations and gender.

The third research question investigates how the different models tested in this study handle undefined gender nouns from the source to the target. Even if LLMs show a more balanced behavior in handling undefined gender, all models demonstrate a preference for outputting male-centric and stereotypical translations, underscoring the challenge of neutrality when gender is unspecified in the source text.

While the present study offers some insights into gender representation in English-to-Portuguese AT, there are several limitations. Firstly, the dataset includes only 120 source sentences, a subset intentionally selected to provide experimental foundations for larger-scale studies focused on the aspects that prove to be most determinant. While this scope is appropriate for a preliminary investigation, it constrains the statistical robustness and generalizability of the findings. Expanding the dataset to include a broader range of professions, sentence structures, and discourse contexts would strengthen future analyses. Secondly, although the annotation of pronoun translations captured two main categories (i.e., recoverable gender in the translation, and unrecoverable gender), a finer-grained classification of translation strategies could provide deeper insight into model behavior, for example, personal pronouns with or without gender marks, null subjects with or without gender-marked part-of-speech, type of possessive constructions, etc. Lastly, the study analyzed marked gender-defined nouns, undefined-gender nouns, and pronoun translations as separate components. However, a more integrated analysis of how these elements correlate could reveal patterns of system consistency or divergence in handling gender related information in translation.

In conclusion, this research demonstrates that in English-Portuguese AT, it is still challenging for the two MT systems (Google Translate and DeepL Translator) and the two LLMs (ChatGPT-4o and DeepSeek-V3) tested to transmit correctly all the gender information from the source to the target text, and that AT outputs lean towards male-centric and stereotypical gender expression, regardless of whether gender is explicitly defined or undefined in the source, especially in the two MT systems.

Future work could design a larger-scale corpus to incorporate more instances of contexts shown to be challenging in this study, and incorporate pronoun referential tracking to capture the interaction among gender cues inside the sentence. Furthermore, as previously noted, even though the English source sentences in WinoMT are structurally ambiguous with respect to pronoun reference, human translators can resolve this ambiguity based on real-world knowledge and pragmatic reasoning to arrive at a contextually appropriate interpretation. Future work could explore whether these limitations can be mitigated through prompt engineering in LLMs, for instance, by providing explicit instructions about pronoun reference resolution, or through few-shot prompting with human-translated examples that demonstrate gender-accurate, context-sensitive translations. Such approaches may offer a practical pathway toward reducing stereotyped default translations in gender-marked target languages in AT.

References

- 1 Sofia Bonifácio, Helena Moniz, and Luisa Coheur. Diving into gender translation bias for the portuguese language. In *28th European Conference on Artificial Intelligence, ECAI 2025, including 14th Conference on Prestigious Applications of Intelligent Systems, PAIS 2025*, pages 1067–1074. IOS Press BV, 2025.
- 2 Won Ik Cho, Ji Won Kim, Seok Min Kim, and Nam Soo Kim. On measuring gender bias in translation of gender-neutral pronouns. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 173–181, 2019.

- 3 Greville G Corbett. *Gender*. Cambridge University Press, 1991.
- 4 Marta R Costa-Jussà, Carlos Escolano, Christine Basta, Javier Ferrando, Roser Batlle, and Ksenia Kharitonova. Interpreting gender bias in neural machine translation: Multilingual architecture matters. In *Proceedings of the aaai conference on artificial intelligence*, volume 36, pages 11855–11863, 2022. doi:10.1609/AAAI.V36I11.21442.
- 5 Eve Fleisig and Christiane Fellbaum. Mitigating gender bias in machine translation through adversarial learning. *arXiv preprint arXiv:2203.10675*, 2022. doi:10.48550/arXiv.2203.10675.
- 6 Joel Escudé Font and Marta R Costa-Jussa. Equalizing gender bias in neural machine translation with word embeddings techniques. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 147–154, 2019.
- 7 Charles F Hockett. *A course in modern linguistics*. The Macmillan Company, 1958.
- 8 Anna Kibort and Greville G Corbett. Grammatical features inventory. *Typology of grammatical features*, 2008.
- 9 Shahar Levy, Koren Lazar, and Gabriel Stanovsky. Collecting a large-scale gender bias dataset for coreference resolution and machine translation. In *Findings of the association for computational linguistics: EMNLP 2021*, pages 2470–2480, 2021. doi:10.18653/V1/2021.FINDINGS-EMNLP.211.
- 10 Maja Popović and Ekaterina Lapshinova-Koltunski. Gender and bias in amazon review translations: by humans, mt systems and chatgpt. In *Proceedings of the 2nd International Workshop on Gender-Inclusive Translation Technologies*, pages 22–30, 2024.
- 11 Marcelo OR Prates, Pedro H Avelar, and Luís C Lamb. Assessing gender bias in machine translation: a case study with google translate. *Neural Computing and Applications*, 32(10):6363–6381, 2020. doi:10.1007/S00521-019-04144-6.
- 12 Randolph Quirk and David Crystal. *A comprehensive grammar of the English language*. Pearson Education India, 2010.
- 13 Eduardo Buzaglo Raposo, Maria Fernanda Nascimento, Maria Antónia Mota, Luísa Segura, and Amália Mendes. *Gramática do Português*. Fundação Calouste Gulbenkian, Lisboa, 2013.
- 14 Aleix Sant, Carlos Escolano, Audrey Mash, Francesca De Luca Fornaciari, and Maite Melero. The power of prompts: Evaluating and mitigating gender bias in mt with llms. In *Proceedings of the 5th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 94–139, 2024.
- 15 Danielle Saunders and Bill Byrne. Reducing gender bias in neural machine translation as a domain adaptation problem. In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pages 7724–7736, 2020. doi:10.18653/V1/2020.ACL-MAIN.690.
- 16 Beatrice Savoldi, Sara Papi, Matteo Negri, Ana Guerberof-Arenas, and Luisa Bentivogli. What the harm? quantifying the tangible impact of gender bias in machine translation with a human-centered study. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18048–18076, 2024. doi:10.18653/V1/2024.EMNLP-MAIN.1002.
- 17 Peter Siemund. *Pronominal gender in English*. Routledge London, 2008.
- 18 Gabriel Stanovsky, Noah A Smith, and Luke Zettlemoyer. Evaluating gender bias in machine translation. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1679–1684, 2019. doi:10.18653/V1/P19-1164.
- 19 Katie Wales. *Personal pronouns in present-day English*. Cambridge University Press, 1996.
- 20 Xiaolan Xu and Sara Mendes. xu-xiaolan/gender-bias-en-pt-at. Dataset, swbId: swb:1:dir:f3d848b15460545b49033fd09dfa2af692d74510 (visited on 2026-07-22). URL: <https://github.com/xu-xiaolan/gender-bias-en-pt-at>, doi:10.4230/artifacts.27322.
- 21 Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2979–2989, 2017. doi:10.18653/V1/D17-1323.

A Occupation nouns tested in this study and their gender-stereotype, as retrieved from WinoMT

Male stereotype occupations: analyst, carpenter, CEO, chief, construction worker, cook, developer, driver, farmer, guard, janitor, lawyer, manager, mechanic, physician, salesperson.

Female stereotype occupations: accountant, assistant, auditor, baker, cashier, clerk, counselor, hairdresser, housekeeper, librarian, nurse, secretary, tailor, teacher, writer.

Cost-Effective Hate Speech Detection in Portuguese Using Lightweight LLMs

Mauro Cardoso 

Instituto Universitário de Lisboa (ISCTE-IUL), Portugal
INESC-ID Lisboa, Portugal

Eugénio Ribeiro 

INESC-ID Lisboa, Portugal
ISTAR, Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

Fernando Batista 

Instituto Universitário de Lisboa (ISCTE-IUL), Portugal
INESC-ID Lisboa, Portugal

Abstract

Addressing Online Hate Speech (OHS) in Portuguese faces significant challenges, including a lack of comprehensive, annotated data and classification frameworks, effective detection tools, and the high computational cost of Large Language Models (LLMs). This work evaluates the efficiency and competitiveness of seven smaller, more accessible LLMs (ranging from 3B to 27B parameters) through zero-shot classification with structured instructions, leveraging the expert-annotated test dataset and annotation scheme of the *kNowHATE* project. Findings indicate that models such as *Mistral Small 3.2 24B*, *Gemma 3 27B*, and *Phi-4* achieve competitive performance, balancing precision and recall, whilst remaining computationally affordable. While excelling in identifying direct and indirect hate speech, out-group derogation, and specific target groups, the models faced challenges in detecting subtle rhetorical devices and emotions. Analysis of the *Cohen's Kappa* and F1-scores revealed moderate agreement with human annotators, highlighting the potential of optimised models to democratise OHS detection in resource-constrained settings, alongside the need for further refinement to bridge the gap in human-level nuance and improve generalisation. This study aims to broaden knowledge regarding the detection of OHS in Portuguese, by demonstrating a viable, competitive, and low-cost approach that does not rely on large-scale infrastructure or expensive proprietary models.

2012 ACM Subject Classification Computing methodologies → Natural language processing; Human-centered computing → Social network analysis; Computing methodologies → Language resources; Applied computing → Sociology

Keywords and phrases Hate Speech Detection, Zero-shot Classification, Lightweight Language Models

Digital Object Identifier 10.4230/OASICS.SLATE.2026.9

Funding This work was supported by Portuguese national funds through Fundação para a Ciência e a Tecnologia, I.P. (FCT) under projects UID/50021/2025 (DOI: <https://doi.org/10.54499/UID/50021/2025>), UID/PRR/50021/2025 (DOI: <https://doi.org/10.54499/UID/PRR/50021/2025>), and UID/04466/2025 (DOI: <https://doi.org/10.54499/UID/04466/2025>).

1 Introduction

The automatic detection of Online Hate Speech (OHS) has become a critical challenge in the age of social media, particularly in multicultural contexts such as Portuguese-speaking communities ([9]). Recent studies highlight the complexity of this multifaceted phenomenon, both in terms of its definition and its manifestation on social media platforms, given that this type of content is predominantly expressed through indirect and subtle language rather



© Mauro Cardoso, Eugénio Ribeiro, and Fernando Batista;
licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 9; pp. 9:1–9:16

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

than overt aggression, and it is often masked by discursive strategies and negative emotions. This type of discourse is primarily aimed at marginalised or minority communities, and has been systematically exploited by political actors as a strategy for polarisation and electoral mobilisation, intensifying social tensions and undermining democratic values ([18, 20, 6, 4, 21]). While Large Language Models (LLMs) have revolutionised the automatic detection of hateful content, they often struggle with slang, vulgar language, or context that may not be inherently toxic, and performance is highly influenced by their capability, purpose, techniques used and how prompts are designed.

Research on hate speech detection in English is well-established [27, 13], however, there's a shortage of publicly available labelled and annotated datasets in Portuguese, which poses a significant challenge to the development, refining, evaluation, and deployment of robust detection systems by general practitioners, limiting the applicability of state-of-the-art models ([26, 5]). Annotated data, being the cornerstone of training and evaluating supervised machine learning algorithms, allows models to capture the semantic nuances and context of human language. In this case, quality takes precedence over quantity: even small datasets, if balanced and carefully labelled by experts, can significantly improve model performance ([31, 7, 26]). Though essential, data annotation can be subjective, as annotators' perceptions vary depending on their identity and experiences ([10]), and success depends not only on the use of robust metrics to evaluate model performance, but also on the influence of annotators in the annotation process ([31, 9, 19]). Despite these challenges, the adoption of LLMs is further constrained by computational and financial costs due to the large number of parameters, and the ongoing need for specialised fine-tuning to handle tasks more efficiently, making their practical application unsustainable, particularly for non-profit organisations, small-scale platforms, or researchers in resource-constrained environments ([5, 2, 9, 1]). Moreover, the scope of analysis in common computational approaches remains limited when compared to the in-depth, systematic examination of OHS provided by specialised studies in Portuguese ([18, 29, 30, 28, 14]). Unlike the prevailing binary approaches, this study adopts the multidimensional labels defined in these works to capture the nuances of hate speech that are frequently overlooked, encompassing discourse types (direct vs. indirect), target communities (racialised groups, migrants, LGBTI+), discrimination processes, rhetorical devices, discursive strategies, fallacies, and emotions. In this context, modern classification methodologies, such as zero-shot classification using structured prompts, are emerging as promising alternatives, enabling the potential and generalisation capabilities of smaller or optimised language models to be exploited, reducing costs and providing a viable alternative with competitive performance ([19, 15, 9, 26, 5, 3]).

This work aims to fill these gaps by extending the analysis and evaluation of OHS components in Portuguese, testing and comparing low-cost language models in a predefined parameters range, offering an accessible and replicable methodology with competitive results. The following research questions are addressed:

- How do small-scale language models, chosen for their low computational resource requirements, perform in detecting OHS components in Portuguese using zero-shot classification with prompts?
- What is the effectiveness and trade-offs of these small-scale language models compared to human annotators?

This document is organised as follows: Section 1 introduces the study, and specifies the scope of the study. Section 2 is a comprehensive review of relevant prior work. Section 3 describes the research methodology, including the data and language models selection,

prompt design, introduces the multi-dimensional classification framework and the evaluation methodologies. Section 4 discusses the results and provides a detailed analysis of the findings. Finally, Section 5 summarises the main findings of the study, discusses its limitations and potential implications, and provides directions for future research.

2 Related Work

This section introduces relevant literature for the study, particularly in the Portuguese context, providing an initial understanding of the context of OHS, and then addressing relevant methodological approaches.

2.1 Understanding Online Hate Speech Content in Portuguese

Based on the *kNOwHATE: kNOwing online HATE speech*¹ project and associated research ([18, 29, 30, 28, 14]), OHS is seen as biased, harmful language that spreads or supports hate and violence against individuals or groups based on their social identity. From a social psychology perspective, OHS is an intergroup issue that reflects bias, often through negative, hostile comments aimed at out-groups. It can also involve seeing the out-group as a threat to the in-group's interests. OHS manifests in two primary forms: direct and indirect expressions, where direct expressions are characterised by explicitly biased and depreciative language, often involving slurs and provocative insults, and in the other hand, indirect hate speech is more common and complex, avoiding explicit depreciatory terms in favour of implicit meanings that must be deduced pragmatically by the audience. OHS often conveys harmful messages through subtle yet effective means, resorting to discursive strategies such as stereotyping, dehumanisation, and the articulation of threats, each reinforcing prejudiced assumptions or instilling fear. Rhetorical devices like irony, sarcasm, metaphors, hyperboles, and rhetorical questions further amplify these messages, often masking their harmful intent while undermining automated detection. Additionally, logical fallacies, such as calls to action or appeals to fear are used to justify hostility or incite harmful measures against targeted groups. Together, these strategies and devices normalise and propagate harmful ideologies within online spaces.

In Portugal, OHS is shaped by social, historical, and cultural factors, where research identifies four main groups targeted: racialised communities (including Afro-descendants and Roma), migrants, and LGBTI+ individuals. The way hate speech manifests against these groups is influenced by Portugal's distinct national narratives: Lusotropicalism, the idea that Portuguese colonisers were uniquely skilled at peaceful intercultural relations, downplaying systemic racism; Anti-Roma Sentiment, centring on stereotypes of criminality, laziness, and supposed resistance to society integration; Sexual Prejudice, targeting LGBTI+ communities through sexual stigma and the perception that they violate traditional in-group values, often triggering expressions of moral disgust and anger. When analysed on Portuguese social media platforms, using expert-annotated datasets (*YouTube*: 24,739 entries; *Twitter/X*: 29,758 entries) and a rigorous annotation process, including inter-annotator agreement measures, results reveal that discriminatory discourse is predominantly indirect, and the primary mechanism of discrimination involves the derogation of out-groups, manifested through stereotyping, threats, and dehumanisation, often amplified by emotions such as hate and anger. Discursive strategies vary according to the targeted community, with irony and

¹ <https://knowhate.eu/>

metaphors emerging as prevalent tactics, LGBTI+ communities are the primary targets on *Twitter/X*, while afrodescendant and roma communities face the most discrimination on *YouTube*. These findings highlight the urgent need to assess the scope and effectiveness of automated detection of OHS in the above contexts.

2.2 Overview on Automated Detection of Online Hate Speech

The automated detection of OHS and toxic content on social media sits at the crossroads of Natural Language Processing (NLP) and computational sociolinguistics. This field has evolved from simple dictionary-based approaches to advanced transformer models and LLMs. This progress has produced a shift from static, lexicon-based methods toward dynamic, context-aware architectures capable of more sophisticated semantic reasoning ([5, 19, 3]).

In this context, while fine-tuned Bidirectional Encoder Representations from Transformers (BERT) models perform well on data similar to their training, LLMs are more resilient to changes in data distribution, where in tests across different datasets, they tend to maintain higher F1-scores than BERT models, which can struggle when faced with unfamiliar data [8, 5]. These models have introduced novel classification capabilities that were largely unattainable with earlier techniques, most notably through zero-shot and few-shot detection of harmful content ([8, 9, 19, 3, 5]). Unlike traditional classifiers that require extensive task-specific training data, like annotated datasets, which is scarce in many languages ([8, 19, 31]), LLMs can identify toxic content by leveraging their broad pre-existing knowledge through natural language instructions, or prompts, opening the possibility of in-context learning, where models adapt to specific linguistic nuances simply by being provided with step-by-step thinking and a few illustrative examples in the input prompt ([2, 8, 5, 9, 19]). Furthermore, LLMs have fundamentally altered the role of the model from a simple classifier to an annotator or expert rationale generator, where research demonstrates that LLMs like *GPT-4* can outperform human crowd-workers in content moderation tasks, such as labelling toxic content, providing a scalable solution to the data scarcity problem in low-resource languages ([19, 3, 4]). Portuguese language relates to this scarcity, despite significant current progress. Even with advanced models, high-quality annotated data remains crucial, as it directly shapes model performance during both training and evaluation phases, and success depends on the precision of these annotations and the use of reliable evaluation metrics ([5, 10, 31]). However, this process depends on trained experts to define hate speech standards, though their judgments can still be influenced by sociocultural and personal biases. For example, an annotator's background can shape perceptions: someone of African descent might interpret aggressive language differently, and female annotators may more often flag gender-based insensitivity than male annotators ([10]).

Despite their cutting-edge status, LLMs are not a universal solution for OHS, requiring far more computational and financial resources than encoder-only models, which can sometimes be more effective for specific tasks, especially on social media ([8, 5]). Performance also depends heavily on prompt design and parametrisation, where harmless text can still be misclassified, particularly when it includes group identifiers or swear words ([8, 19]). This is where LLMs with smaller parameters come into play, typically ranging from 3B parameters upward, and within the zero-shot learning paradigm, they present a complex trade-off between computational efficiency and classification robustness. Though, the field has favoured large proprietary decoder models, recent studies show that compact, often open-source models can match their performance in toxic content detection, especially when fine-tuned or guided with optimised prompts for domain-specific tasks ([9, 19, 3]).

2.3 Zero-Shot Classification for Hate Speech Detection in Portuguese

Zero-shot classification looks promising for detecting hate speech in Portuguese, especially when interpretability and adaptability matter. Still, its practical use must account for computational demands, pointing toward smaller models as a way to balance effectiveness, cost, and efficiency.

Large scale decoder models as well as optimised models such as *GPT-4o* mini, demonstrate high resilience to data distribution shifts, often outperforming fine-tuned encoder models like *BERTimbau* in cross-dataset scenarios, offering sophisticated understanding to offensive content, providing natural language rationales for their decisions ([8, 9, 19]). Other smaller or optimised models, such as *LLaMA 3.1 8B*, *Phi-3-Mini (3.8B)*, and *Falcon (7B)*, offer a more accessible alternative for local deployment. While *LLaMA 3.1 8B* in its original zero-shot state may show lower F1-scores compared to larger models, it achieves high recall, although at the cost of precision ([8, 9]). When optimised through iterative prompt refinement, these 8B models can match or surpass the precision of encoder models like *BERTimbau*, and the performance of *GPT-4o* mini.

Studies show that the effectiveness of zero-shot classification depends on two key layers of prompts: the instruction and the context. A well-structured prompt typically integrates three core elements – system instruction, which defines the model’s role and task boundaries; Contextual enhancement, which provides relevant keywords to guide multilingual models; Task specificity, where a balance must be struck between narrowly focused prompts (which yield high recall but lower precision) and broader prompts (which better balance precision and recall by including terms like aggression or offence) ([8, 9, 5, 2]). Template design often involves simple formats like “This text is” or “Classify this text as [option 1], [option 2], or [option 3].”, a binary response “yes” or “no.”, or advanced setups combining system messages (e.g., “You are performing text analysis”) with user messages that include the specific question and the target text ([26, 8, 5]). For instructional styles, one approach is the *Vanilla* method, which directly asks whether content belongs to a class (e.g., “Is the following comment hateful?”). Another method is Chain-of-Thought (CoT), requiring the model to reason step-by-step before providing a label. A further technique is Role-Play, where the model is assigned a specific identity, such as a community moderator. Additionally, Definition-based prompts, which provide formal definitions of target classes to reduce ambiguity [15, 19, 2]. To accurately evaluate OHS detection in Portuguese, a robust approach is essential, one that accounts for the language’s subjectivity and the scarcity of large annotated datasets. Performance is primarily measured using macro-averaged F1-scores to ensure fair comparison across imbalanced classes, while precision and recall help reveal biases ([8, 15]). Standardised frameworks, datasets and metrics such as *Cohen’s Kappa*, *Fleiss’ Kappa*, and *Krippendorff’s alpha*, provide consistent benchmarks and inter-rater reliability, testing robustness through controlled cases involving implicit hate, negation, and non-hateful slurs. ([15, 19, 8, 5, 31, 18]).

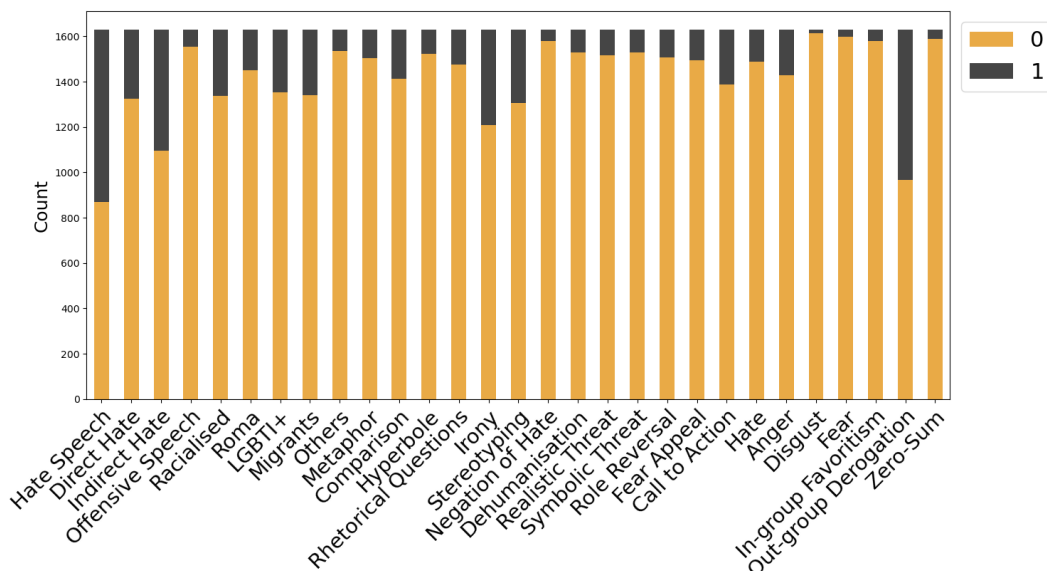
3 Research Methodology

Having established the theoretical and practical challenges of OHS detection in Portuguese, this section outlines the data, models selection and prompt design rationale, the multi-dimensional framework, and lastly, the evaluation methodology.

3.1 Data Selection

As introduced in Section 2, this study uses the *Twitter/X* and *Youtube* test dataset made available by the project *kNOwHATE: kNOwing online HATE speech* ([17, 18, 29, 30, 28, 14]). The full dataset, collected between 2021 and 2022, comprises approximately 37,000 geolocated tweets originating from Portugal. It was gathered using a lexical-based approach, guided by a predefined list of terms tied to target communities, and to refine the dataset, it was applied a filtering strategy for ambiguous terms; for example, the word “preto” (which can mean “black” or serve as a slur) was only retained when it appeared alongside offensive or derogatory language. The dataset was meticulously annotated by a trained interdisciplinary team of social psychologists and linguists, who assessed each comment across 47 variables. These included the type of speech (offensive, direct or indirect hate speech,), targets (specific marginalized communities), discrimination processes (motivations such as out-group derogation or zero-sum perceptions), discursive and rhetorical features (stereotypes, dehumanization, threats, irony, metaphors, and logical fallacies like call to action or appeal to fear), and emotions (hate, anger, disgust, or fear). All variables were binary and non-mutually exclusive. Annotation reliability, measured using *Krippendorff’s alpha*, revealed stronger agreement for objective categories like target identification than for subjective categories like subtle irony or specific emotions.

Due to the terms of the grant agreement with the European Union, it has been classified as sensitive and is therefore not publicly accessible. However, researchers can access a subset of the this golden set used for experimental test, via the Open Science Framework (OSF) ([17]), resulting in a total of 1630 posts with 62 columns and 47 labels. Figure 1 presents the distribution of positive (1) and negative (0) values of 29 labels in the context of OHS, filtered for classification and analysis ([18]), where the definitions of these labels will be explained in Section 3.3. The visualisation highlights a marked under-representation of positive label classes, initially suggesting significant challenges for classification and evaluation tasks.



■ **Figure 1** Selected Labels Distribution of the Twitter(X) and YouTube test data.

3.2 Language Model Selection and Activity Overview

A wide range of language models is now available, creating a dynamic and diverse ecosystem with varying strategies, efficiency levels, and specialisations. From this ecosystem, seven models were chosen based on their accessibility on OpenRouter, technical characteristics, provider uptime performance, reputation of the model’s owner, and most importantly, the low cost. Table 1 presents some metrics related to the models selected and an overview of model usage within this study, in which we highlight the diverse parameter scale coverage among the selected language models. This distribution highlights the availability of both compact and mid-sized models, which are particularly relevant for resource-constrained environments, where the proprietary models (*Gemini 2.0 Flash Lite*) stands out from the open source remaining models, due to its undisclosed parameter count, suggesting a potential trade-off between performance and transparency. While input/output pricing varies (as is common in the industry, where outputs are often more expensive than inputs), all selected models allowed for cost optimisation depending on the provider, with highly competitive rates, at no more than \$0.08 per million input tokens and no more than \$0.30 per million output tokens. Parameter sizes range from 3B to 27B, influencing both capability and cost. Request volumes are relatively consistent across models, ranging from 47k to 60k per 1M tokens. *Llama 3.2 3B Instruct* emerges as the most economical, and *Gemma 3 27B* the most expensive. Models like *Gemini 2.0 Flash Lite* and *Mistral Small 3.2* offer a balance between cost and advanced features, while *Phi-4* is constrained by a shorter context window.

■ **Table 1** Language models selected and activity overview for online hate speech detection based on cost efficiency, context length, and model capacity.

Model	Parameters	Context Len.	I/O Price*	Cost, Tokens, Requests
Google: Gemini 2.0 Flash Lite ([16])	NA	1.05M	\$0.075/\$0.30	\$0.410, 2.54M, 48k
Google: Gemma 3 27B ([11])	27B	131k	\$0.08/\$0.16	\$1.126, 6.18M, 50k
Meta: Llama 3.1 8B Instruct ([22])	8B	131k	\$0.016/\$0.021	\$0.161, 6.69M, 47k
Meta: Llama 3.2 3B Instruct ([23])	3B	80k	\$0.003/\$0.006	\$0.348, 6.74M, 48k
Mistral: Mistral Small 3.2 24B ([25])	24B	131k	\$0.075/\$0.20	\$0.466, 5.82M, 48k
Mistral: Mistral Nemo ([24])	12B	128k	\$0.02/\$0.04	\$0.1268, 5.79M, 47k
Microsoft: Phi-4 ([12])	14B	16.4	\$0.065/\$0.14	\$0.5276, 8.14M, 60k

*Price per 1M tokens for input/output, depending on provider (OpenRouter, April 2026).

3.3 Multi-Dimensional Classification Framework

In order to expand and evaluate the ability of models to detect and analyse OHS components in Portuguese, it was necessary to establish a robust framework. This framework is based on the *kNOwHATE: kNOwING online HATE speech* project ([18, 29, 30, 28, 14, 17]), which provides both an annotation scheme and comprehensive guidelines that are central to this study. These elements form the theoretical and methodological foundation for the classification system employed in the selected models, where Tables 2 and 3 show this framework, which is based on a curated selection of 29 labels, aiming to simplify the scope of the classification approach.

These labels were derived from multiple analytical dimensions and definitions, as specified in the annotation scheme and refined through subsequent research [18, 17]. Each label is accompanied by its guiding questions and operational definitions, illustrating how the classification framework systematically integrates target groups, rhetorical devices, hate speech types, emotional markers, and discursive strategies into a cohesive analytical structure.

■ **Table 2** Multi-Dimensional Framework for Zero-Shot Hate Speech Classification: Hate Speech Types, Target Groups, Rhetorical Devices, and Fallacies. (Part I).

Label	Question	Definition
<i>Hate Speech Types</i>		
Hate Speech	Does this text contain hate speech?	Speech that contains incitement, insults or discrimination based on social identity, whether direct or indirect.
Direct Hate	Does this text express Direct Hate Speech?	Explicit and aggressive expression of hate against a group based on its social identity.
Indirect Hate	Does this text express Indirect Hate Speech?	Implicit or covert expression of hate, using subtleties or irony to avoid direct accusation.
Offensive Speech	Does this text express Offensive Speech (not necessarily hate)?	Aggressive or insulting language with an individual focus, not necessarily based on social group.
<i>Target Groups</i>		
Racialised	Does this text mention or focus on racialised communities?	Hate speech targeting groups in processes of racialisation, attributing a constructed racial identity that often leads to social exclusion.
Roma	Does this text mention or focus on the Roma (gypsy) community?	Hate speech targeting racialised groups identified through the stigmatising term “gypsy” or similar.
Migrants	Does this text mention or focus on migrants?	Hate speech targeting individuals due to their nationality or migratory origin. Often related to xenophobia.
LGBTI+	Does this text mention or focus on LGBTI+ people?	Hate speech targeting sexual orientation, gender identity/expression, or sex characteristics. Encompasses homophobia, biphobia, transphobia, and intersexphobia.
Others	Does this text mention or focus on other minority groups outside the main categories (Racialised, Roma, Migrants, LGBTI+)?	Hate speech directed at groups or communities not explicitly included in the other categories (Racialised, Roma, Migrants, LGBTI+).
<i>Rhetorical Devices</i>		
Metaphor	Does this text use the rhetorical device of Metaphor?	Transfer of attributes between distinct concepts with the aim of devaluing or dehumanising the target group.
Comparison	Does this text use the rhetorical device of Comparison?	Association of a group with another through analogies or comparisons, aiming to depreciate or devalue.
Hyperbole	Does this text use the rhetorical device of Hyperbole?	Use of intentional exaggeration to provoke emotional reactions or reinforce negative perceptions of the target group.
Rhetorical Questions	Does this text use Rhetorical Questions?	Questions that do not require an answer, but aim to reinforce a negative opinion or stereotype.
Irony	Does this text use the rhetorical device of Irony?	Indirect and ambiguous expression used to mock or humiliate, often in the form of sarcasm or humour.
<i>Fallacies</i>		
Fear Appeal	Does this text use the rhetorical device of Fear Appeal?	Strategy that uses fear to persuade, suggesting negative outcomes if no action is taken.
Call to Action	Does this text use the rhetorical device Call to Action?	Explicit or implicit summons to act against the target group or a situation portrayed as negative.

■ **Table 3** Multi-Dimensional Framework for Zero-Shot Hate Speech Classification: Discursive Strategies, Discrimination, and Emotions (Part II).

Label	Question	Definition
<i>Discursive Strategies</i>		
Stereotyping	Does this text use Stereotyping?	Attribution of fixed and negative characteristics to a group based on incorrect generalisations.
Negation of Hate	Does this text use the strategy of Negation of Hate?	Strategy of presenting one's own discourse as legitimate and non-hateful, even when transgressing social norms.
Dehumanisation	Does this text use Dehumanisation?	Representation of the target group as less human, facilitating hostility or violence against it.
Realistic Threat	Does this text use the strategy of Realistic Threat?	Perception of concrete threat to the resources, security, or well-being of the dominant group.
Symbolic Threat	Does this text use the strategy of Symbolic Threat?	Perception of threat to the values, beliefs, or identity of the dominant group.
Role Reversal	Does this text use Role Reversal?	Presenting the dominant group as the victim and the marginalised group as the aggressor.
<i>Discrimination</i>		
In-group favouritism	Does this text describe the form of discrimination known as in-group favouritism (an explicit preference for members of one's own group)?	An explicit preference for members of one's own group, even at the expense of impartiality.
Out-group Derogation	Does this text describe the form of discrimination known as 'out-group derogation' (the devaluation or hostility towards members of other groups)?	Belittling or hostility directed at external groups.
Zero-Sum	Does this text reflect a form of zero-sum discrimination (favouring one's own group and disparaging the out-group)?	An explicit preference for members of their own group and hostility towards external groups.
<i>Emotions</i>		
Hate	Does this text express the emotion of Hate?	Intense aversion and desire to eliminate or harm a group perceived as evil or immutable.
Anger	Does this text express Anger?	Negative emotion motivated by perception of injustice, often with a desire for confrontation or change.
Disgust	Does this text express Disgust?	Moral or physical rejection of a group, often associated with ideas of contamination or devaluation.
Fear	Does this text express Fear?	Emotion associated with the perception of threat or danger (physical or symbolic) posed by the target group.
Guilt	Does this text express Guilt?	Recognition of moral responsibility for transgressions committed by the in-group.

This framework is especially valuable in zero or few shot classification tasks, where models must generalise from definitions and examples to identify OHS across diverse contexts, without prior exposure to specific instances. Such an approach directly supports the goals of initiatives like the kNOWHATE project, which seeks to develop culturally sensitive and effective methods for OHS detection and prevention, particularly within the Portuguese linguistic and cultural context.

3.4 Classification and Evaluation Approach

For this classification approach, prompts were carefully designed to determine whether a given comment contained a single label related to OHS, grounded in prior research, which shows that model performance tends to decline as the number of classes increases ([3]). The prompts followed a question–statement pair logic, where each text was submitted to a model with responses restricted to binary options – “Yes” or “No” to minimize ambiguity.

Each prompt consisted of three key components: a direct question to guide the model’s analysis, a clear definition outlining the criteria, and the text for analysis. An example of the structure is given as follows:

Prompt Design Components

Question: “Does this text mention or focus on ..?”
Definition: “Hate speech targeting groups in processes of .. ”
Text for analysis: “I hate this people... ”

Algorithm 1 shows the classification process applied, where it involves iterating over each dataset and text instance, while querying each language model with the prompt components to generate the predictions along with the relevant evaluation metrics. The metrics are F1-score, which harmonises precision and recall, false positive (FP) and false negative (FN) rates, and *Cohen’s Kappa* which measures agreement beyond chance (Models Prediction vs Human Annotation).

■ **Algorithm 1** Classification Approach for Hate Speech Detection.

Require:

- 1: Datasets $D = D_1, D_2, \dots, D_n$ with annotations for hate speech.
- 2: Models $L = L_1, L_2, \dots, L_m$ to evaluate.

Ensure:

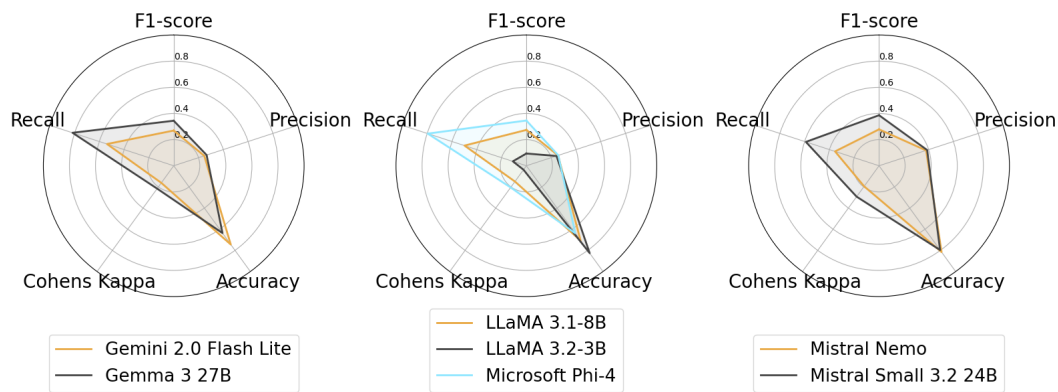
- 3: Performance scores for each model on each dataset.
- 4: **for** each dataset $D_i \in D$ **do**
- 5: **for** each text instance $t \in D_i$ **do**
- 6: **for** each model $L_j \in L$ **do**
- 7: Preprocess text if necessary.
- 8: Create a prompt for model L_j .
- 9: Send prompt to model L_j and obtain response r_{ij} .
- 10: Parse and validate r_{ij} to get predicted label p_{ij} .
- 11: Compare true labels with predictions.
- 12: Compute metrics:
- 13: *Accuracy, Precision, Recall, F1-score, Cohen’s Kappa, TP, FP, TN, FN.*
- 14: **end for**
- 15: **end for**
- 16: **end for**
- 17: Join results.

4 Result Analysis

The results presented in this section are based on the methodology and multidimensional classification framework established in the previous sections, providing a comprehensive evaluation and analysis of the lightweight language models selected for the detection of OHS in Portuguese.

4.1 Models Performance Analysis

The interpretation of the results demands a cautious approach, given the unbalanced nature of the test data and the diversity of models/labels evaluated. The radar charts in Figure 2 provides the starting point for the analysis, comparing the performance of the models selected across five metrics: accuracy, precision, recall, *Cohen's Kappa*, and F1-score. The display is divided into three groups, each with its own legend showing the respective models, to make the analysis clearer.

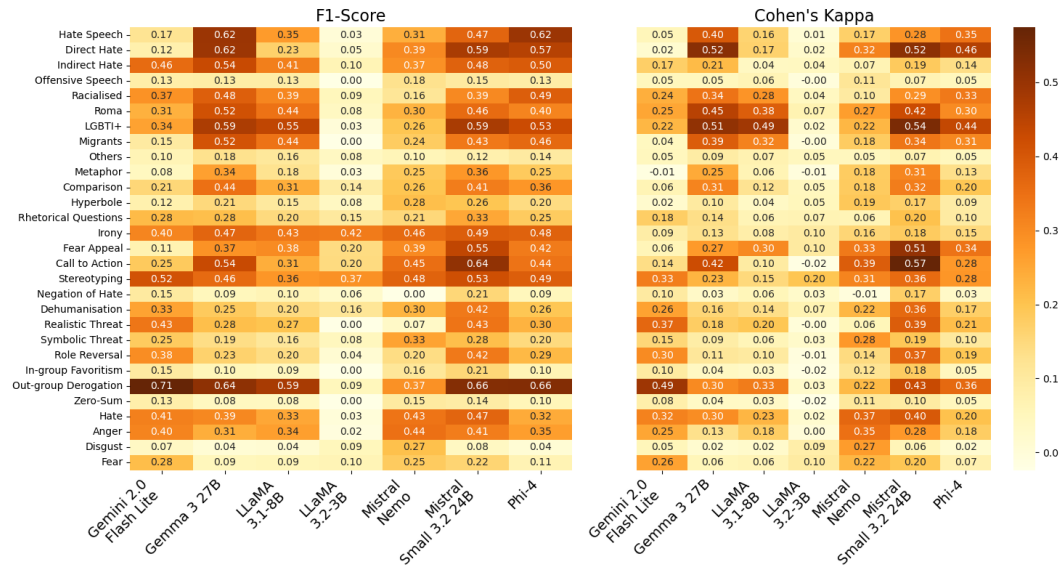


■ **Figure 2** Overview of the models' average performance.

Overall, the models exhibit similar behaviour: although they are able to identify the majority of cases and maintain a good overall accuracy rate and recall, they suffer from a lack of precision. *LLaMA 3.2-3B*, for example, has the highest overall accuracy rate, but is the least balanced and consistent. *Mistral Small 3.2 24B*, on the other hand, stands out as the most robust, delivering a much more balanced performance than other equally robust competitors such as *Gemma 3 27B*, *Mistral Nemo* and *Phi-4*.

The heat maps in Figure 3 present a deeper comparative analysis of F1-score and *Cohen's Kappa* metrics, revealing distinct patterns in the models' ability to detect OHS. The *LLaMA-3.2-3B* model consistently shows lower F1-scores across most labels, particularly for positive instances, despite moderate results in "Stereotyping" (0.37) and "Irony" (0.42). As the most affordable but smallest model, its limited performance suggests it cannot fully capture complex OHS nuances without further refinement. Consequently, applying a strict exclusion criterion, where models failing to achieve an F1-score above 0.15 across more than 70% of labels are discarded to avoid sub-baseline noise. This model is excluded from further analysis in favour of better-performing alternatives.

The *Mistral Small 3.2 24B*, *Gemma 3 27B* and *Phi-4* models stand out for their superior performance in both metrics. Specifically, they achieve high F1-scores for labels such as "LGBTI+", "Out-group Derogation", and "Direct Hate Speech", indicating a robust global capacity for detecting hateful content. Although these models imply higher costs, their



■ **Figure 3** F1-score and *Cohen's Kappa* heatmap for each model by OHS labels.

performance is considered justified and proportional to the complexity and scale of the information processed. The *Gemini 2.0 Flash Lite*, despite its extensive context window, shows only moderate performance relative to the other models. This observation suggests that an excessive context window may not be critical for this specific task, as the model does not significantly outperform others. A notable pattern emerges in the models' ability to generalise across various labels, though difficulties persist in handling less frequent nuances or subclasses, such as "Denial of Hate", "Disgust", "Zero-Sum", and "In-group Favouritism". Additionally, the models exhibit challenges in interpreting labels like "Fear", "Metaphor", "Hyperbole", and "Offensive Speech", which are often camouflaged forms of OHS.

Despite these limitations, the models show a robust capacity for identifying direct and indirect hate speech, aligning with findings from previous research in the field. The F1-scores across models range from moderate to high, while *Cohen's Kappa* values tend to be lower. This discrepancy suggest that, although the models can effectively detect OHS categories, inter-annotator agreement or internal consistency may be constrained. Such a trend underscores the complexity of achieving both high accuracy and reliable human-model agreement in this domain. For applications prioritising cost-effectiveness, the *Mistral Nemo*, *Gemini 2.0 Flash Lite*, and *Llama 3.1 8B* models emerge as the most balanced options, although they would likely require some fine-tuning. These models are particularly suitable for scaling to large datasets without substantially compromising performance, making them ideal for projects with more budget constraints. For scenarios demanding higher performance – within a controlled budget – the *Mistral Small 3.2 24B*, *Gemma 3 27B* and *Phi-4* models stand out as a valid choice. Although they require a higher budget for scaling, the results demonstrate significant value, justifying the investment for tasks where precision and reliability are paramount. Given that, in this case, fine-tuning would likely yield high-quality results. Considering the confusion matrix shown in Figure 4, which presents even more detailed analysis of the top-performing models, *Gemma 3 27B* and *Phi-4* show high sensitivity, detecting a large number of positive cases (true positives).

Label	Mistral Small 3.2 24B				Mistral Nemo				Google Gemma 3 27B				Microsoft Phi-4			
	TP	FP	TN	FN	TP	FP	TN	FN	TP	FP	TN	FN	TP	FP	TN	FN
Hate Speech	248	45	823	514	144	22	846	618	388	104	764	374	422	183	685	340
Direct Hate	155	62	1262	151	84	40	1284	222	199	141	1183	107	193	177	1147	113
Indirect Hate	285	367	727	251	201	338	756	335	388	516	578	148	360	555	539	176
Offensive Speech	70	804	749	7	37	292	1261	40	75	972	581	2	71	911	642	6
Racialised	92	88	1250	200	30	43	1295	262	173	251	1087	119	202	336	1002	90
Roma	54	4	1447	125	34	13	1438	145	105	120	1331	74	109	254	1197	70
Migrants	99	74	1265	192	45	34	1305	246	181	229	1110	110	181	310	1029	110
LGBTI+	122	13	1341	154	43	7	1347	233	155	96	1258	121	153	143	1211	123
Others	11	75	1459	85	9	76	1458	87	47	375	1159	49	50	545	989	46
Metaphor	43	69	1434	84	35	118	1385	92	109	405	1098	18	104	607	896	23
Comparison	90	127	1286	127	45	81	1332	172	168	372	1041	49	174	572	841	43
Hyperbole	94	514	1009	13	84	415	1108	23	103	788	735	4	99	790	733	8
Rhetorical Questions	134	534	942	20	101	708	768	53	148	770	706	6	146	889	587	8
Irony	370	722	485	53	317	639	568	106	404	893	314	19	401	851	356	22
Fear Appeal	89	95	1398	48	53	83	1410	84	119	386	1107	18	98	230	1263	39
Call to Action	151	81	1306	92	81	38	1349	162	206	313	1074	37	206	495	892	37
Stereotyping	246	363	941	80	192	287	1017	134	297	673	631	29	283	553	751	43
Negation of Hate	20	118	1461	31	0	6	1573	51	51	1034	545	0	51	1030	549	0
Dehumanisation	85	216	1313	16	70	297	1232	31	98	592	937	3	89	495	1034	12
Realistic Threat	55	83	1432	60	4	3	1512	111	99	503	1012	16	89	387	1128	26
Symbolic Threat	89	450	1077	14	49	141	1386	54	97	802	725	6	98	756	771	5
Role Reversal	56	86	1421	67	20	61	1446	103	101	670	837	22	86	382	1125	37
In-group Favouritism	13	60	1520	37	16	130	1450	34	37	691	889	13	39	669	911	11
Out-group Derogation	423	203	763	241	165	52	914	499	521	446	520	143	507	368	598	157
Zero-Sum	36	441	1147	6	27	296	1292	15	40	879	709	2	41	764	824	1
Hate	111	219	1270	30	78	141	1348	63	130	392	1097	11	130	549	940	11
Anger	185	506	922	17	108	177	1251	94	200	876	552	2	195	731	697	7
Disgust	16	381	1233	0	3	3	1611	13	16	847	767	0	16	834	780	0
Fear	14	82	1516	18	14	68	1530	18	27	535	1063	5	28	465	1133	4

■ **Figure 4** Top performant models: Confusion matrix metrics by label.

However, this broad coverage results in a clear tendency towards over-classification, as evidenced by the high rates of false positives in almost all categories. In contrast, the Mistral family is showing a more cautious approach, *Mistral Small 3.2 24B* and the *Mistral Nemo* offer high accuracy, with a very low rate of false positive errors compared to other models. The cost of this selectivity is an increase in missed cases (false negatives), particularly in complex categories such as “Out-group Derogation” and “Hate Speech”. Finally, categories that require a nuanced reading between the lines, such as “Irony” and “Indirect Hate Speech”, prove to be a common challenge. In these cases, all models struggle to distinguish the underlying intent, resulting in a delicate balance between correct and incorrect classifications.

5 Conclusions and Future Work

This study suggests that lightweight language models ranging from 12B to 27B parameters, such as *Mistral Small 3.2 24B*, *Gemma 3 27B* and *Phi-4*, serve as computationally efficient and cost-effective alternatives for detecting OHS in Portuguese. By using a zero-shot classification approach within a multi-dimensional framework adapted from the *kNOwHATE* project, these models achieve competitive performance, finding a necessary balance between classification robustness and operational affordability.

Regarding the first research question (RQ1), the findings shows that smaller-scale models can achieve competitive performance for OHS detection in low resource environments ([19, 9]), particularly when combined with techniques like *zero-shot* classification optimised through prompt engineering and structured definitions. While these models excel at identifying direct and indirect hate speech and specific target groups, such as *LGBTI+*, *Racialised communities*, and *Roma*, the detection of nuanced rhetorical devices – including irony, hyperbole, and metaphor – remains a persistent challenge. This difficulty often leads to over-classification and a higher rate of false positives, which highlights the ongoing need for more in-depth semantic reasoning to accurately analyse the underlying intent. In relation to the second research question (RQ2), the alignment between model outputs and human annotations, as measured by *Cohen’s Kappa*, reveals a nuanced performance landscape. Across most labels, the models achieved moderate agreement with the expert-annotated set, though consistently below the inter-annotator agreement observed among human experts. Specifically, for example models in general demonstrated relatively high recall and accuracy, aligning well with the majority of expert annotations for direct and indirect hate speech, out-group derogation, and target group identification. However, their precision lagged behind human annotators, particularly for ambiguous or culturally specific expressions.

Several limitations must be acknowledged, notably the reliance on specific sample datasets and the presence of class imbalance, which may have skewed performance metrics by inflating recall while masking precision deficiencies. Furthermore, the zero-shot paradigm lacks the domain-specific fine-tuning necessary to adapt to evolving slang and the dynamic nature of hateful discourse. Not including different and adaptive suggestion techniques, and the constraints of shorter context windows may also have limited the models’ ability to capture long-range dependencies in more complex textual structures.

Future research should prioritise enhancing dataset representativeness and exploring hybrid strategies that integrate zero/few-shot and fine-tuning approaches. To achieve this, it is essential to address model bias and deploy these frameworks in diverse environments, including political contexts and community-led monitoring, thereby ensuring more ethical, transparent, and culturally sensitive OHS detection solutions for the Portuguese language.

References

- 1 Aish Albladi, Minarul Islam, Amit Das, Maryam Bigonah, Zheng Zhang, Fatemeh Jamshidi, Mostafa Rahgouy, Nilanjana Raychawdhary, Daniela Marghitu, and Cheryl Seals. Hate Speech Detection Using Large Language Models: A Comprehensive Review. *IEEE Access*, 13:20871–20892, 2025. doi:10.1109/ACCESS.2025.3532397.
- 2 Gabriel Assis, Annie Amorim, Jonnathan Carvalho, Daniel de Oliveira, Daniela Vianna, and Aline Paes. Exploring Portuguese Hate Speech Detection in Low-Resource Settings: Lightly Tuning Encoder Models or In-Context Learning of Large Models? In *Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1*, pages 301–311. Association for Computational Linguistics, 2024. URL: <https://aclanthology.org/2024.propor-1.31/>.
- 3 Basel Barakat and Sardar Jaf. Beyond Traditional Classifiers: Evaluating Large Language Models for Robust Hate Speech Detection. *Computation*, 13(8), 2025. doi:10.3390/computation13080196.
- 4 Mauro Cardoso, Eugénio Ribeiro, and Fernando Batista. Portuguese Far-Right Discourse on Social Media: Insights from Topic Modeling. In *14th Symposium on Languages, Applications and Technologies (SLATE 2025)*, volume 135 of *Open Access Series in Informatics (OASICS)*, pages 12:1–12:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/OASICS.SLATE.2025.12.
- 5 Amanda et al. da Silva Oliveira. Toxic Speech Detection in Portuguese: A Comparative Study of Large Language Models. In *Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1*, pages 108–116. Association for Computational Linguistics, 2024. URL: <https://aclanthology.org/2024.propor-1.11/>.
- 6 Thomas Davidson, Dana Warmesley, Michael Macy, and Ingmar Weber. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*, pages 512–515, 2017. doi:10.1609/icwsml.v11i1.14955.
- 7 J. Angel Diaz-Garcia and Joao Paulo Carvalho. A Literature Review of Textual Cyber Abuse Detection Using Cutting-Edge Natural Language Processing Techniques: Language Models and Large Language Models. *WIREs Data Mining and Knowledge Discovery*, 15(3):e70029, 2025. doi:10.1002/widm.70029.
- 8 Amanda Oliveira et al. How Good Is ChatGPT For Detecting Hate Speech In Portuguese? In *Anais do XIV Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 94–103, Porto Alegre, RS, Brasil, 2023. SBC. doi:10.5753/stil.2023.233943.
- 9 Amanda Oliveira et al. Toxic Text Classification in Portuguese: Is LLaMA 3.1 8B All You Need? In *Anais do XV Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 57–66. SBC, 2024. doi:10.5753/stil.2024.245416.
- 10 Amit Das et al. Investigating Annotator Bias in Large Language Models for Hate Speech Detection. *ArXiv*, 2024. doi:10.48550/arXiv.2406.11109.
- 11 Gemma Team et al. Gemma 3 technical report, 2025. doi:10.48550/arXiv.2503.19786.
- 12 Microsoft et. al. Phi-4 technical report. *arXiv*, 2024. arXiv:2412.08905v1.
- 13 Zahra Safdari Fesaghandis and Suman Kalyan Maity. Multilingual hate speech detection and counterspeech generation: A comprehensive survey and practical guide. *arXiv preprint arXiv:2603.19279*, 2026. doi:10.48550/arXiv.2603.19279.
- 14 Pedro Fialho, Ricardo Ribeiro, Fernando Batista, Gil Ramos, António Fonseca, Sérgio Moro, Rita Guerra, Paula Carvalho, Catarina Marques, and Cláudia Silva. *Counter Hate Speech Detection in Youtube Conversations*, pages 94–105. Springer Nature Switzerland, 2025. doi:10.1007/978-3-031-73997-2_9.
- 15 Faeze Ghorbanpour, Daryna Dementieva, and Alexander Fraser. Can Prompting LLMs Unlock Hate Speech Detection across Languages? A Zero-shot and Few-shot Study. In *Proceedings of the The 9th Workshop on Online Abuse and Harms (WOAH)*, pages 413–425, Vienna, Austria, August 2025. Association for Computational Linguistics. URL: <https://aclanthology.org/2025.woah-1.39/>.

- 16 Google DeepMind. Gemini 2.0 flash lite: Documentation and technical overview. <https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-0-flash-lite>, 2024. Access: April 2026.
- 17 Rita Guerra, Fernando Batista, Paula Carvalho, and Ricardo Ribeiro. Data (Golden set/Test files), November 2025. doi:10.17605/OSF.IO/MPEJ3.
- 18 Rita Guerra, Paula Carvalho, Catarina Marques, Margarida Carmona, Rodrigo Sarroeira, Fernando Batista, Ricardo Ribeiro, António Fonseca, Sérgio Moro, and Cláudia Silva. Unpacking online hate speech in Portuguese social media: a social-psychological and linguistic-discursive approach. *Humanities and Social Sciences Communications*, 12:1709, 2025. doi:10.1057/s41599-025-05392-9.
- 19 Tharindu Kumarage, Amrita Bhattacharjee, and Joshua Garland. Harnessing Artificial Intelligence to Combat Online Hate: Exploring the Challenges and Opportunities of Large Language Models in Hate Speech Detection. *arXiv*, 2024. doi:10.48550/arXiv.2403.08035.
- 20 Tiago Lapa and Branco di Fátima. Hate Speech Among Security Forces in Portugal. *Communication Library*, 7:277–293, 2023. doi:10.25768/654-916-9.
- 21 Luca Manucci. Portuguese Populism: People, Parties, and Politics. *Análise Social*, 59(2 (251)):2–11, 2024. doi:10.31447/202200.
- 22 Meta. Llama 3.1: Model cards & prompt formats. https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_1/, 2024. Access: April 2026.
- 23 Meta. Llama 3.2: Model cards & prompt formats. https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_2/, 2024. Access: April 2026.
- 24 Mistral AI. Mistral-nemo: Architecture and performance report. <https://mistral.ai/news/mistral-nemo>, 2024. Access: April 2026.
- 25 Mistral AI. Mistral 3: Release notes and benchmarks. <https://mistral.ai/news/mistral-small-3>, 2025. Access: April 2026.
- 26 Flor Miriam Plaza-del arco, Debora Nozza, and Dirk Hovy. Respectful or Toxic? Using Zero-Shot Learning with Language Models to Detect Hate Speech. In *The 7th Workshop on Online Abuse and Harms (WOAH)*, pages 60–68. Association for Computational Linguistics, 2023. doi:10.18653/v1/2023.woah-1.6.
- 27 Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, 55:477–523, June 2021. doi:10.1007/s10579-020-09502-8.
- 28 Catarina Pontes, António Fonseca, Sérgio Moro, Fernando Batista, Ricardo Ribeiro, Catarina Marques, Paula Carvalho, Cláudia Silva, and Rita Guerra. *Unveiling Patterns of Hate Speech in the Portuguese Sphere: A Social Network Analysis Approach*, pages 70–81. Springer Nature Switzerland, 2025. doi:10.1007/978-3-031-73997-2_7.
- 29 Gil Ramos, Fernando Batista, Ricardo Ribeiro, Pedro Fialho, Sérgio Moro, António Fonseca, Rita Guerra, Paula Carvalho, Catarina Marques, and Cláudia Silva. A comprehensive review on automatic hate speech detection in the age of the transformer. *Social Network Analysis and Mining*, 14:204, 2024. doi:10.1007/s13278-024-01361-3.
- 30 Gil Ramos, Fernando Batista, Ricardo Ribeiro, Pedro Fialho, Sérgio Moro, António Fonseca, Rita Guerra, Paula Carvalho, Catarina Marques, and Cláudia Silva. Leveraging Transfer Learning for Hate Speech Detection in Portuguese Social Media Posts. *IEEE Access*, 12:101374–101389, 2024. doi:10.1109/ACCESS.2024.3430848.
- 31 Natalia Umansky, Maël Kubli, Ana Kotarcic, Laura Bronner, Selina Kurer, Philip Grech, Dominik Hangartner, Fabrizio Gilardi, and Karsten Donnay. Improving hate speech detection with large language models. *European Journal of Political Research*, pages 1–12, January 2026. doi:10.1017/S1475676525100546.

Assessing LLMs for Culturally Sensitive Communication in European Portuguese

Alice Vieira ✉ 

CLUNL – Research Center for Linguistics at NOVA University Lisbon,
NOVA University Lisbon, Portugal

Raquel Amaro ✉ 

CLUNL – Research Center for Linguistics at NOVA University Lisbon,
NOVA University Lisbon, Portugal

Lyndon Nixon ✉ 

Storypact GmbH, Vienna, Austria

Abstract

This paper investigates the ability of LLMs to identify culturally sensitive communication in European Portuguese, in comparison with human performance. The study addresses cross-cultural communication issues arising within a single language, including miscommunication, bias, framing, and stereotyping. The task is formulated as a multi-component problem combining detection, classification, span identification, and reformulation of problematic content. A dataset of 60 sentences, balanced between problematic and neutral instances, was constructed from real-world corpora and annotated by expert linguists. Three open-weight multilingual models were evaluated alongside human annotators with diverse backgrounds. Results show that LLMs achieve high accuracy in detecting problematic content, outperforming non-expert participants and approaching expert performance. However, both models and humans exhibit low agreement in the classification of communication issues, reflecting the conceptual overlap between categories. While models generate consistent reformulations, they show limitations in span precision and contextual interpretation. The findings highlight a gap between detection and interpretation capabilities and support the use of LLMs as assistive tools in cross-cultural communication tasks, particularly in politically sensitive contexts.

2012 ACM Subject Classification Computing methodologies → Natural language processing; Computing methodologies → Language resources; Computing methodologies → Machine learning; Human-centered computing → Human computer interaction (HCI)

Keywords and phrases LLM, culturally sensitive communication, evaluation

Digital Object Identifier 10.4230/OASICS.SLATE.2026.10

Funding *Alice Vieira*: European Commission (Project MultiPoD, ref. 101178821, <https://multipod-project.eu/>), and Portuguese national funds through FCT (CLUNL, DOI:10.54499/UID/03213/2025). *Raquel Amaro*: European Commission (Project MultiPoD, ref. 101178821, <https://multipod-project.eu/>), and Portuguese national funds through FCT (CLUNL, DOI:10.54499/UID/03213/2025). *Lyndon Nixon*: European Commission (Project MultiPoD, ref. 101178821, <https://multipod-project.eu/>).

1 Introduction

In political deliberation processes, both deliberate and unintended communication barriers may arise either during translation or within the same language, potentially hindering effective interaction [9]. Even within the same language, the meaning and interpretation of words can be mediated by the sociocultural context, political views and educational or vocational background of both the speaker and the listener [28, 26, 3]. While considerable effort is devoted to ensuring accuracy in machine translation, particularly in preserving



© Alice Vieira, Raquel Amaro, and Lyndon Nixon;
licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 10; pp. 10:1–10:14
OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

meaning between source and target languages [16], there exists a tendency to assume that no further “translation” is required within a single language. That is, it is often assumed that communication issues are solved when two communicators share the same language. However, linguistic research has demonstrated that speakers of the same language may not necessarily share the same semantic or pragmatic interpretations [17, 11]. For instance, a British English speaker instructing an American English speaker to “put out a fag” refers to a cigarette, whereas in North American usage the term “fag” carries a derogatory meaning. In this example, the listener may interpret the statement as offensive, leading to defensive or confrontational communication dynamics.

1.1 The MultiPoD project

The work presented here is situated within the MultiPoD project¹, whose objective is to align communication between participants in a multilingual space toward a shared understanding, thereby reducing barriers to effective communication and enhancing democratic participation and political deliberation. While misalignments in cross-lingual communication may primarily result from incorrect or ambiguous translation, cross-cultural communication issues often arise between interlocutors using the same language, whether natively or through translation. Such issues typically originate in differences between the speaker’s intended meaning and the listener’s interpretation, including cases of miscommunication, misinterpretation, and misunderstanding [28, 27].

A range of issues may emerge in cross-cultural communication, and communication studies and sociolinguistics have proposed various classifications of problematic or sensitive language contributing to it. These commonly include categories such as insult, provocation, prejudice or bias, stereotyping or discrimination, framing, cultural misunderstanding, including pragmatic failure, cultural insensitivity, and misogyny or gender bias [31, 8, 29]. Although no single standardized taxonomy exists, and individual cases may fall into multiple categories or be subject to disagreement among human interpreters, such classification frameworks are useful for ensuring comprehensive coverage of cross-cultural communication phenomena. In particular, they should help capture more implicit forms of communication barriers that arise from cultural context rather than explicit use of offensive language.

The present study constitutes an exploratory step towards this broader objective, focusing on the evaluation of LLM performance in identifying and reformulating culturally sensitive communication phenomena in European Portuguese. Given the pragmatic and sociocultural complexity of such phenomena, the operationalisation of annotation categories presents important methodological challenges. In particular, certain categories may partially overlap in practice, and both human annotators and language models may legitimately diverge in their interpretation of specific cases. Rather than assuming the existence of fully discrete or universally agreed-upon labels, the study examines how these challenges affect the computational treatment of cross-cultural communication.

1.2 Culturally sensitive language model: goal and research question

To address these challenges computationally, a fine-tuned language model for cross-cultural understanding is being developed in MultiPoD, supported by a culture-specific knowledge graph that encodes lexical and conceptual knowledge associated with specific sociocultural,

¹ <https://multipod-project.eu>

political, and economic contexts. The integration of structured knowledge with language models has been shown to improve contextual understanding and interpretability in NLP systems [14, 22]. In this framework, both the language model and the knowledge graph are made available to cross-cultural communication processes, supporting functionalities such as content transformation, summarization, and recommendation. This approach aims to enable the detection, interpretation, and mitigation of communication barriers before these adversely affect interaction between participants. In the context of political deliberation, such tools have the potential to support smoother communication processes and, consequently, more effective deliberation outcomes.

A culturally-sensitive language model is defined as being able to identify potential cross-cultural misunderstandings and to propose alternatives that support effective cross-cultural communication. The model is expected to evaluate multiple dimensions of linguistic appropriateness, including the use of polite and respectful language, the use of gender-inclusive expressions, the avoidance of stereotypes and generalizations, particularly in relation to the “people-first” principle, the use of accessible and plain language, minimizing reliance on specialized or technical terminology, and the avoidance of implicit cultural norms or cultural meanings while allowing for cultural diversity, [26, 12, 2].

Within the project, and focusing on European Portuguese, the goal of the work presented here is to assess whether existing models can support culturally sensitive communication in European Portuguese, achieving over 90% accuracy in the detection of cross-cultural communication problems and over 80% relevance in the suggested reformulations of the text, as defined in the project proposal.

2 Related work

Recent research has focused on culture and LLMs, including determining if models are culturally sensitive and how culturally sensitive models they are. Such research has shown that performance cannot be reduced to language performance or multilingual coverage. Model outputs are shaped by culturally uneven training data, prompting language, and the evaluation framework itself.

Concerning cultural values and cultural alignment in LLMs, Leet et al. [18] argue that mainstream LLMs are biased toward cultures overrepresented in English-dominant training corpora and propose a cost-effective fine-tuning approach based on World Values Survey (WVS) data [13] plus semantic data augmentation. Their experiments cover 9 cultures, including Portuguese, and show that culture-specific adaptation can improve performance on downstream culture-related tasks such as offensive language, hate, toxicity, stance, and bias detection. However, they work on a multilingual setting covering Brazil and parts of Latin America, and do not address European Portuguese. Bulte and Terryn [4] provide an explicit comparison between model outputs and human cultural values. Using 10 LLMs, 63 items from the Hofstede Values Survey Module [10] and the WVS, and prompts translated into 11 languages, the authors show that both prompt language and cultural framing affect model responses. Their results suggest that prompting in a local language may change responses, but in existing LLMs that does not necessarily make responses more human-like or culturally faithful. Related evidence also appears in the context of cultural bias mitigation. For instance, Narayan et al. [21], compare a general model with a model incrementally trained on Black history and culture, showing that targeted cultural enrichment can reduce racial and cultural bias in generated summaries. The paper supports the broader claim that culturally grounded data and evaluation are necessary if LLMs are to be trusted in socially sensitive communicative settings.

Focusing on multilingual and community-specific adaptation, El Mekki et al. [7] point out that adaptation pipelines that rely only on machine translation from English may transfer the source culture together with the source text. They propose a combined strategy using translation, controlled synthetic generation with local personas, and retrieval of community-specific cultural knowledge. Their proof of concept for Egyptian and Moroccan Arabic shows that adaptation should jointly consider language, cultural heritage, and cultural values, but use external fixed resources to do so. This concern is echoed by Van Doren and Holland [30], where in a human evaluation study of 87 translations across 24 dialects of 20 languages, they show that even when LLM outputs are grammatically acceptable, figurative language, idioms, puns, and audience-sensitive localization often remain problematic and require human revision. Although covering Portuguese, they focus on translation and cover Brazilian and not European varieties. However, their work demonstrates that human assessment is indispensable to ensure cultural appropriateness and communicative adequacy.

A complementary perspective is provided by Luo [20], who proposes a Cross-Cultural Adaptation Framework for multilingual LLMs. This framework explicitly models cultural adaptation as a multi-stage process involving (i) cultural context detection, (ii) knowledge integration, (iii) adaptive response generation, and (iv) cultural evaluation. The architecture organizes these components into a pipeline where cultural information is progressively incorporated into the generation process. Empirical results show substantial improvements in cultural appropriateness (around 91% compared to 62–83% for baselines), highlighting that systematic integration of cultural knowledge outperforms translation-only or prompt-based approaches. Importantly, cultural adaptation must operate at lexical, pragmatic, and contextual levels and evaluation requires dedicated metrics capturing cultural appropriateness rather than relying solely on generic NLP metrics.

The role of human comparison and human expertise is also addressed in recent research. Aleem et al. [1] show that generic LLMs can fail in culturally nuanced interactions because of limited adaptability, weak cultural humility, and insufficient understanding of user perspective. Building on this gap, Liu [19] introduce the CultureCare dataset and compare adapted LLM responses with human peer responses using LLM-as-a-Judge, in-culture human annotators, and clinical psychologists. Although their domain is emotional support, their methods and results show that culturally sensitive communication benefits from comparison with human responses.

Several studies focus on harmful, biased, or implicit language, considering these as highly relevant to cross-cultural communication because many communicative failures are indirect. Cheremetiev et al. [5] propose specializing general-purpose LLM embeddings to address implicit hate speech, which operates through subtle cues, sarcasm, stereotypes, and context-dependent meanings. Likewise, Cheriyan et al. [6] treat offensive language as communication that can be harmful on moral, social, religious, or cultural grounds and propose not only detection but also reformulation for conflict reduction. These studies motivate the need to include indirect, pragmatic phenomena, not just clearly offensive language, as part of cross-cultural communication problems.

Related work shows that LLMs are consistently affected by cultural bias and often default to dominant perspectives. However, multilingual prompting or translation does not guarantee on its own culturally appropriate communication. The evaluation of the systems requires also human judgment, especially in tasks involving nuance, reformulation, implicit meaning, and social acceptability. Nonetheless, the setting targeted in the present paper, namely cross-cultural communication in the same language, and the direct comparison between human and language models performance on detecting and reformulating culturally problematic communication is still not addressed.

3 Methodology and experimental setup

This section describes the methodology and experimental setup used, namely in what concerns task design, evaluation data, interaction protocols, and evaluation metrics.

3.1 Task definition

As our study focuses on comparing multilingual LLMs and human performance in supporting culturally sensitive communication, the task was formulated as a multi-component problem combining detection, classification, span identification, and reformulation.

Given an input sentence, both models and human annotators were required to:

- determine whether the sentence contains a cross-cultural communication issue (binary classification),
- identify the specific textual span(s) responsible for the issue,
- assign one or more labels describing the type of communication problem.
- generate an alternative formulation that mitigates the issue while preserving the original meaning.

This formulation aligns with prior work on text simplification and controlled text generation, where both semantic fidelity and pragmatic appropriateness are essential dimensions of performance [32, 24].

3.2 LLMs and human evaluation

3.2.1 Model selection

Model selection was guided by three main criteria: (i) multilingual capability with support for Portuguese, (ii) computational feasibility for local execution and fine-tuning, and (iii) availability within a reproducible deployment environment. The selection also focused on medium-sized language models that can be executed, and, if necessary, fine-tuned, within a reasonable computational time frame. In addition to computational efficiency, a key requirement was that the models be available as open-source or open-weight implementations, thereby enabling reproducibility and potential adaptation to the specific requirements of the task.

Model selection was further constrained to models available for loading and inference via non-proprietary libraries or APIs so that we could re-use the same evaluation code rather than build a new evaluation pipeline for every new model. We looked at Ollama and HuggingFace, which both allow model downloading onto a local machine for inference, with preference given to models already adopted across application domains and actively maintained. This ensures both practical usability and long-term sustainability.

Based on these criteria, the following models were selected: `mistral-7b-portuguese`², OpenEuroLLM Portuguese³, and `gervasio:8b`⁴.

² https://ollama.com/cnmoro/mistral_7b_portuguese

³ <https://ollama.com/jobautomation/OpenEuroLLM-Portuguese>

⁴ <https://huggingface.co/PORTULAN/gervasio-8b-portuguese-ptpt-decoder>

3.2.2 Human evaluators

The group of human evaluators consisted of seven participants recruited from university staff and the researchers’ wider network, ensuring diversity in background, as detailed in Figure 1. The group included three experts in Linguistics, alongside contributors from other disciplinary areas, supporting a range of interpretative perspectives. This diversity is particularly relevant in cross-cultural communication, where sensitivity to pragmatic meaning, implicature, and sociocultural norms may vary across individuals [28, 11, 27]. In

Expertise/Professional background	Age	Gender	Proficiency
Linguistics	20-30	Man	Native speaker
Linguistics	30-40	Man	Native Speaker
Linguistics	30-40	Female	Highly Proficient
Data Science	20-30	Man	Native Speaker
Literature	40-50	Female	Native Speaker
Design	40-50	Man	Native Speaker
Physics	50-60	Female	Native Speaker

■ **Figure 1** Sociodemographic information of human evaluators.

addition to disciplinary diversity, the evaluator group also included participants with lived experience connected to communities that are frequently affected by bias in automated and AI-mediated communication systems, including members of LGBTQ+ communities and individuals from immigrant and racialized backgrounds. While these characteristics were not formally operationalized as variables in the study, their presence contributed to a broader range of interpretative perspectives during the evaluation process, particularly regarding sensitivity to sociocultural nuance, representation, and potential bias in generated content.

3.3 Experimental setup

3.3.1 Category set

The set of categories considered as potentially culturally sensitive are presented in Figure 2. These categories were based on existing literature, and their definitions were designed both for human interpretation and LLM processing. Although certain categories may appear conceptually similar, such as “Insult” and “Provocation”, important distinctions remain between them. For instance, the sentence *If you do not support this measure, it is because you live in a completely different world.* would be categorized as Provocation rather than Insult, since it does not exhibit the salient ad hominem features that typically characterize insults.

Label	Short definition
Framing	Language that shapes the perception of information by selectively highlighting or omitting specific details
Stereotyping	Language that expresses assumptions about people based on their belonging to a group, reducing individuals to generalized characteristics
Miscommunication	Unclear or ambiguous language that causes the message to be received differently from what was intended
Insult	Language used with the purpose of attacking or offending people
Misunderstanding	Language that can be misinterpreted, creating confusion about the speaker’s intentions or the expected response
Provocation	Language intended to provoke someone or cause a negative reaction
Bias	Language that favors one group, idea, or perspective over others

■ **Figure 2** Categories used in the task.

3.3.2 Evaluation Data

The evaluation dataset consists of 60 sentences: 30 containing the cross-cultural communication issues considered, and 30 neutral sentences (i.e., without such issues). Ensuring a balanced representation of problematic and “null” cases is essential for reliable evaluation [23, 15].

Constructing a data set of sentences exemplifying cross-cultural communication issues, along with their classification, annotated spans, and neutral reformulations, represents several methodological challenges. Approaches based solely on expert-generated data do not scale effectively [25] and may fail to capture the full range of implicit pragmatic and sociocultural phenomena. To mitigate these limitations, real-world examples were collected from the Portuguese Web 2023⁵ (European Portuguese subcorpus) and the MIGRANTE.PT corpus⁶, and subsequently validated, classified and annotated by native European Portuguese linguists.

The categorization framework adopted in this study was established during a preliminary analytical phase conducted by the researchers, drawing on existing literature on cross-cultural communication and pragmatic interpretation. This process resulted in the predefined category set presented in Figure 2, as well as the reference annotations associated with the evaluation dataset. The subsequent evaluation phase did not seek to generate categories inductively; rather, both human evaluators and LLMs were asked to identify, classify, and reformulate instances according to the predefined framework. Consequently, the comparison evaluates the extent to which human participants and LLMs aligned with the reference categorization and annotations established during dataset construction.

3.3.3 Interaction protocol

The human evaluation followed a multi-task protocol combining binary classification (neutral vs. problematic), categorization of communication issues, span identification, and reformulation of problematic content. Each participant analyzed the same evaluation set of 60 sentences. For each sentence, participants were asked to:

- indicate whether it was neutral or problematic;
- if problematic, assign one or more categories based on the predefined labels (Figure 2);
- identify the specific word(s) or phrase(s) conveying the problematic content (span);
- provide a neutral reformulation of the identified text.

The annotation task was conducted without prior training, using only brief instructions and label definitions. To minimize ordering effects, sentences were presented in randomized order. The task was administered online using the Gorilla platform⁷ between November and December 2025.

LLMs were prompted to process each input sentence and produce outputs in a structured JSONL format, consistent with the reference annotations, as indicated in Figure 3.

This format includes classification (problematic vs. non-problematic), span identification, category labels, and reformulated text. The use of a structured output format enabled systematic and programmatic comparison between model predictions and human annotations.

⁵ <https://www.sketchengine.eu/pttnten-portuguese-corpus/>

⁶ <https://clunl.fcsh.unl.pt/recursos-em-linha/corpora/migrante-pt/>

⁷ <https://gorilla.sc/>

You have the role of an interpreter for cross-cultural communication.

Your task is to identify any portion of a sentence which is culturally problematic due to being in the category of:

- Insult (use of strong, insulting language)
- Provocation (use of language deliberately to provoke the listener)
- Bias (use of language which exhibits a social, cultural or political bias)
- Stereotyping (use of language which stereotypes a social, cultural or political group)
- Framing (use of language to emphasize one aspect of a topic while downplaying or omitting other, equally relevant, aspects)
- Misunderstanding (use of language by the speaker which can be incorrectly interpreted by the listener due to different backgrounds or contexts of interpretation)
- Miscommunication (use of unclear or ambiguous language which may not be understood by the listener in the intended manner)

You will highlight those portions, label them with the category they belong to and suggest an alternative wording which is not problematic. The response to every sentence is one line of JSONL markup which repeats the text of the sentence and 0 or more annotations of the problematic parts. Example(s):

```
{
  "text": "A imigração é a solução para substituir os portugueses.",
  "span": "substituir os portugueses",
  "label": "Framing",
  "alternative": "colmatar a falta de população ativa"
}
```

■ **Figure 3** Example of the prompt and JSONL format.

3.4 Evaluation metrics

Task design facilitated programmatic analysis of responses in comparison with the original evaluation dataset. Performance was evaluated using multiple complementary metrics, each capturing a different aspect of the task:

- Accuracy score based on the binary classification of each sentence (problematic/not problematic)
- Token-level recall score (span detection) on the overlap between the reference and annotated spans
- Accuracy of the classification of the type of cross-cultural communication issue
- Semantic similarity measure (reformulation quality) between the reference alternative text and the generated reformulations.

The evaluation of reformulations follows approaches used in text simplification and bias mitigation, where both semantic fidelity and pragmatic appropriateness are considered essential.

4 Results

4.1 Human evaluation

The results of the human evaluation task are summarized in Figure 4 and cover three main dimensions: binary classification accuracy, categorization agreement (measuring the degree to which participants assigned the same categories as the reference annotations), and span detection.

Inter-rater agreement for the binary classification task was calculated using Fleiss' K. Expert participants achieved substantial agreement ($K = 0.744$, $p < 0.001$), whereas non-expert participants achieved fair agreement ($K = 0.232$, $p < 0.05$). These results further reflect the subjective nature of the task and indicate that participants with relevant expertise showed greater consistency in the identification of culturally sensitive content.

Participants	Classification*	Categorization	Span detection**
Experts	80% Precision 1.0 Recall 0.8	43,3% 16 different category 13 same category 6 marked as neutral	partial correspondence: 15 exact correspondence: 7 no consensus: 4 not marked: 4 no correspondence: 0
Non-experts	36,7% Precision 1.0 Recall 0.7		partial correspondence: 16 exact correspondence: 8 no consensus: 4 not marked: 2 no correspondence: 0

*Calculated per majority

**Span detection is analyzed as: *exact correspondence*; *partial correspondence*, comprising cases of original span + more and only part of original span; *no correspondence*; *no consensus*, where exact and partial correspondences were marked in equal proportion (no majority); and *not marked*, where no part of the sentence was marked by the participants

■ **Figure 4** Summary of human evaluation results.

Experts achieved a classification accuracy of 80%, with precision = 1.0 and recall = 0.8, indicating that all sentences identified as problematic were indeed problematic, while some problematic cases were not detected. In contrast, non-experts obtained a substantially lower accuracy of 36.7%, although precision remained at 1.0 and recall at 0.7, suggesting a conservative strategy with frequent omission of problematic cases.

Categorization results reveal considerable variability. Among expert annotations, 16 cases were assigned different categories, 13 matched the reference category, and 6 were marked as neutral, indicating disagreement not only in category selection but also in the identification of problematic content. This variability confirms the difficulty of assigning fine-grained categories to cross-cultural communication issues. These categories were based on existing literature.

Span detection shows relatively consistent behaviour: experts achieved 7 exact matches and 15 partial matches, with 4 cases of no consensus and 4 instances where no span was marked; non-experts produced 8 exact matches and 16 partial matches, with slightly fewer unmarked cases. No instances of completely incorrect span identification (no correspondence) were observed, suggesting that once an issue is detected, participants are generally able to localize it with some degree of accuracy.

The reformulation did not allow for systematic comparison, due to high variability in participant strategies. Some participants reformulated entire sentences, while others focused only on the identified span or proposed multiple alternatives. In several cases, participants explicitly stated that reformulation was not possible without additional context. A total of six sentences were classified as problematic contrary to the reference annotations. Four of these instances were identified by the same non-expert participant, who considered the absence of contextual information as a source of ambiguity or bias (e.g., “Programmes to support the middle class were approved.” was suggested to be reformulated as “Programmes to support the middle class in area X were approved.”). The remaining two cases were identified by an expert participant, who interpreted adjectives and adverbs as introducing subjectivity, thereby compromising the neutrality of the statements (e.g., “Tourism contributes significantly to the national economy” was reformulated as “Tourism is one of the drivers of the national economy.”).

4.2 LLM evaluation

The results presented in Figure 5 reveal notable variation in model performance across the evaluated dimensions. Every model was loaded onto a local machine (laptop with a single GPU) and prompted with the same explanatory prompt for each individual sentence

Model	Binary classification	Span detection	Label classification	Quality of alternative text
mistral-7b-portuguese	78.33%	0.10 Recall 0.10	1%	43.54%
OpenEuroLLM Portuguese	91.67%	0.55 Recall 0.72	8%	64.02%
gervasio-8b	86.67%	0.65 Recall 0.71	5%	53.14%

■ **Figure 5** Summary of model evaluation results.

in the evaluation dataset (outlining the task, an one sentence definition of each type of cross-culturally problematic communication, ten examples as part of few-shot prompting, and a specification of the JSONL format it should use in its response). The full prompt is passed every time to ensure deterministic, repeatable model behavior avoiding increasing ambiguity and drift.

Models achieved relatively high scores in the binary classification of problematic/non-problematic, with OpenEuroLLM Portuguese showing the strongest performance (91.67%), indicating a solid ability to distinguish between problematic and non-problematic content. Span detection (selection of the culturally sensitive part of the sentence) results are more variable. gervasio-8b achieves the highest score (0.65, recall 0.71), closely followed by OpenEuroLLM Portuguese (0.55, recall 0.72), while mistral-7b-portuguese performs poorly (0.10). This indicates that identifying the precise location of problematic content remains challenging for some models. Label classification performance is consistently low (1%-8%), confirming that fine-grained categorization of cross-cultural communication issues is the most difficult component of the task.

Reformulation quality is comparatively stronger, measured by using SentenceBERT to calculate the semantic similarity of the model reformulation to the gold standard of the evaluation data. OpenEuroLLM Portuguese reaching 64.02%, followed by gervasio-8b (53.14%) and mistral-7b-portuguese (43.54%). This suggests that models can generate acceptable alternative formulations, although the values are never very high indicating that models are unlikely to follow precisely a single “gold standard” option.

Overall, the results indicate a clear performance gap between issue detection and issue interpretation, with models performing well in the former but struggling with precise categorization and span identification.

4.3 Discussion

The results obtained from both automated and human evaluation highlight the inherently subjective and context-dependent nature of culturally sensitive communication.

It is possible to observe that, in this experiment, LLMs outperform non-expert participants in binary classification, and in some cases approach expert-level performance. For example, OpenEuroLLM Portuguese (91.67%) exceeds expert accuracy (80%) in identifying problematic sentences. However, this advantage is limited to detection: both humans and models show low agreement in category assignment, indicating that classification is inherently unstable across annotators and systems.

Span detection results suggest complementary strengths. Humans, particularly experts, show higher precision in identifying relevant segments, with no cases of completely incorrect spans. LLMs, while achieving comparable recall (≈ 0.7), show more variability in precision, especially in smaller models such as mistral-7b-portuguese. This indicates that models can detect relevant areas but often lack fine-grained control in delimiting them.

The very low label classification accuracy (1%–8%) contrasts sharply with human disagreement patterns, where even experts diverge in 16 out of 35 cases. This suggests that the difficulty lies not only in model limitations but also in the lack of clear boundaries between categories, particularly for phenomena such as framing, bias, and misunderstanding, which often overlap in practice.

Another factor that may have contributed to variability in human annotation concerns the decision to conduct the task without extensive prior training, relying instead on brief instructions and category definitions. Given the pragmatic and context-dependent nature of cross-cultural communication phenomena, additional calibration sessions or iterative annotation rounds might have improved consistency between annotators, particularly for categories with partial conceptual overlap. At the same time, the absence of intensive training reflects more natural interpretative conditions, in which individuals rely on their own sociocultural and pragmatic understanding when evaluating potentially sensitive communication. The results therefore highlight both the difficulty of the annotation task itself and the methodological challenges involved in designing sufficiently robust and consistently interpretable category frameworks.

Finally, reformulation results highlight a divergence between procedural and interpretative capabilities. Models produce relatively consistent reformulations (up to 64.02%), whereas human outputs are more diverse and less comparable. However, human reformulations often reflect deeper contextual reasoning, as seen in cases where participants introduced missing information or adjusted evaluative tone. This indicates that model-generated alternatives are structurally adequate but may lack contextual grounding.

The variability in human judgements, as well as the diversity and creativity of human-generated reformulations, point to the challenges of formalizing such phenomena within discrete categories, on the one hand, and of fully automating their resolution, on the other.

Overall, the findings further suggest that the evaluation data set itself provides a conservative benchmark for assessing model performance, as the real-world-grounded, expert-validated reference baseline captures a broader and more inclusive range of potentially problematic cases. This is reflected in both human disagreement and model errors, particularly in sentences involving implicit bias or contextual ambiguity.

5 Final remarks and future work

This exploratory study provides a comparative evaluation of human and LLM performance on a multi-component task involving detection, classification, span identification, and reformulation of cross-cultural communication issues in European Portuguese. The results show that current LLMs are effective at identifying problematic content, achieving classification accuracy above 85%, remain limited in fine-grained analysis, with low performance in label classification and inconsistent span detection. These limitations are particularly evident when compared to expert annotations, which, despite internal disagreement, demonstrate more precise localization and context-sensitive interpretation.

At the same time, human evaluation reveals substantial variability, especially in categorization and reformulation strategies. The fact that experts disagreed in nearly half of the categorization cases indicates not only variability in interpretation but also a broader methodological challenge in operationalizing cross-cultural communication phenomena into discrete annotation categories. Categories such as insult, provocation, framing, or cultural insensitivity may partially overlap in practice, particularly in context-dependent or pragmatically ambiguous cases. This has direct implications for model evaluation, as it challenges the assumption of a single “correct” annotation and suggests that disagreement may reflect the inherent complexity of the task rather than annotation failure alone.

Taken together, these findings indicate that cross-cultural communication analysis requires both detection and interpretation mechanisms, and that current LLMs primarily address the former. As a result, they are better suited for flagging potentially problematic content than for providing reliable explanations or categorizations. Consequently, the integration of automated detection mechanisms with human validation emerges as a key design principle for systems operating in deliberative or participatory environments.

In this context, our future work will focus on three main directions: (i) redefining the category set, reducing overlap between labels and improving annotation consistency; (ii) improving detection of subtle communication issues and accurate span detection by LLMs through fine-tuning with the culture-specific data; and (iii) ensuring that AI-suggested reformulations are always optional for human users and need their consent for rewriting culturally sensitive statements. Our next evaluation for cross-cultural sensitivity detection by LLMs will use an expanded data set, include more varied and subtle contexts (e.g., such as value mismatch and cultural drift) and add more languages, which will be essential to ensure evaluation of both model performance and their generalization to other languages.

From a broader perspective, this work also contributes to ongoing research on the interaction between linguistic structures, semantic interpretation, and computational modeling. By grounding language models in validated, culture-specific language data, it becomes possible to move towards a more context-aware understanding of language use. This is a starting point for mediation of cross-cultural communication and related areas such as discourse analysis, multilingual NLP, and human–computer interaction for European Portuguese and, by extension, other languages and their regional or cultural variants.

References

- 1 Muhammad Aleem, Iqra Zahoor, and Muhammad Naseem. Towards culturally adaptive large language models in mental health: Using chatgpt as a case study. In *Companion of the 2024 Computer-Supported Cooperative Work and Social Computing (CSCW Companion '24)*. ACM, 2024. doi:10.1145/3678884.3681858.
- 2 Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 610–623. ACM, 2021. doi:10.1145/3442188.3445922.
- 3 Jan Blommaert. *The Sociolinguistics of Globalization*. Cambridge University Press, 2010.
- 4 Bram Bulte and Ayla Rigouts Terryn. Llms and cultural values: The impact of prompt language and explicit cultural framing. *Computational Linguistics*, 2026. doi:10.1162/COLI.a.583.
- 5 Vassiliy Cheremetiev, Quang Long Ho Ngo, Chau Ying Kot, Alina Elena Baia, and Andrea Cavallaro. Specializing general-purpose llm embeddings for implicit hate speech detection across datasets. In *Proceedings of the ACM Workshop on Detection of Harmful Online Content*, 2025.
- 6 Jithin Cheriyan, Bastin Tony Roy Savarimuthu, and Stephen Cranefield. Towards offensive language detection and reduction in four software engineering communities. In *Proceedings of the Evaluation and Assessment in Software Engineering (EASE 2021)*, pages 254–259. ACM, 2021. doi:10.1145/3463274.3463805.
- 7 Abdelkader El Mekki, Hiba Atou, Onur Nacar, Samer Shehata, and Muhammad Abdul-Mageed. Nilechat: Towards linguistically diverse and culturally aware llms for local communities. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP 2025)*, pages 10967–10991, 2025.
- 8 Norman Fairclough. *Critical Discourse Analysis: The Critical Study of Language*. Routledge, 2nd edition, 2013.
- 9 Jürgen Habermas. *The Theory of Communicative Action*, volume 1. Beacon Press, 1984.

- 10 Geert Hofstede and Michael Minkov. Values survey module 2013, 2013. Accessed: 2026-05-19. URL: <https://geerthofstede.com/research-and-vsm/vsm-2013/>.
- 11 Juliane House. *Translation Quality Assessment: Past and Present*. Routledge, 2015.
- 12 Dirk Hovy and Shannon L. Spruit. The social impact of natural language processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 591–598. ACL, 2016. doi:10.18653/v1/P16-2096.
- 13 Ronald Inglehart, Christian Haerpfer, Alejandro Moreno, Christian Welzel, Kseniya Kizilova, Juan Diez-Medrano, Marta Lagos, Pippa Norris, Eduard Ponarin, and Bi Puranen. World values survey: Round six - country-pooled datafile, 2014. Version available at <https://www.worldvaluessurvey.org/WVSDocumentationWV6.jsp>.
- 14 Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 33(2):494–514, 2021. doi:10.1109/TKDE.2019.2940204.
- 15 Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Pearson, 3rd edition, 2023. Draft version available online.
- 16 Philipp Koehn. *Neural Machine Translation*. Cambridge University Press, 2020.
- 17 Stephen C. Levinson. *Pragmatics*. Cambridge University Press, 1983.
- 18 Cheng Li, Ming Chen, Jiahui Wang, Sunayana Sitaram, and Xianchao Xie. Culturellm: Incorporating cultural differences into large language models. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024)*, 2024.
- 19 Chen Cecilia Liu, Hiba Arnaout, Nils Kovačić, Dana Atzil-Slonim, and Iryna Gurevych. Tailored emotional llm-supporter: Enhancing cultural sensitivity. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2026)*, pages 535–574, 2026.
- 20 Xiaofei Luo. Cross-cultural adaptation framework for enhancing large language model outputs in multilingual contexts. *Journal of Advanced Computing Systems*, 3(5):48–62, 2023. doi:10.69987/JACS.2023.30505.
- 21 Meera Narayan, Jonathan Pasmore, Eduardo Sampaio, Vignesh Raghavan, Subhajit Maity, and George Waters. Mitigating bias in large language models through culturally-relevant llms. In *Proceedings of the 2025 IEEE International Symposium on Ethics in Engineering, Science, and Technology (ETHICS 2025)*, 2025. doi:10.1109/ETHICS65148.2025.11098204.
- 22 Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander Miller, and Sebastian Riedel. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2463–2473. ACL, 2019. doi:10.18653/v1/D19-1250.
- 23 Timo Schick and Hinrich Schütze. It’s not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021)*, pages 2339–2352. Association for Computational Linguistics, 2021. doi:10.18653/v1/2021.naacl-main.185.
- 24 Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3407–3412. ACL, 2019. doi:10.18653/v1/D19-1339.
- 25 Rion Snow, Brendan O’Connor, Daniel Jurafsky, and Andrew Ng. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In Mirella Lapata and Hwee Tou Ng, editors, *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. URL: <https://aclanthology.org/D08-1027/>.
- 26 Helen Spencer-Oatey. *Culturally Speaking: Culture, Communication and Politeness Theory*. Continuum, 2nd edition, 2008.

- 27 Helen Spencer-Oatey and Peter Franklin. *Intercultural Interaction: A Multidisciplinary Approach to Intercultural Communication*. Palgrave Macmillan, 2009.
- 28 Jenny Thomas. Cross-cultural pragmatic failure. *Applied Linguistics*, 4(2):91–112, 1983. doi:10.1093/applin/4.2.91.
- 29 Teun A. van Dijk. *Discourse and Power*. Palgrave Macmillan, 2008.
- 30 Max Van Doren and Claire Holland. "be my cheese?": Assessing cultural nuance in multilingual llm translations, 2025. Appen report.
- 31 Ruth Wodak. The discourse-historical approach. In Ruth Wodak and Michael Meyer, editors, *Methods of Critical Discourse Analysis*, pages 63–94. Sage, 2001.
- 32 Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415, 2016. doi:10.1162/tac1_a_00107.

Developing an AI-Based Application for Linguistic Analysis of Deaf Learners' L2 Writing

Aryane Santos Nogueira 

UNICAMP, Campinas-SP, Brasil

Ana Célia Ribeiro Bizigato Portes 

UNICAMP, Campinas-SP, Brasil

FATEC, Mogi Mirim-SP, Brasil

Abstract

The teaching of written language as a second language (L2) to deaf students remains a critical challenge in bilingual education. This is also the case for written Portuguese in Brazil, where difficulties persist particularly in the analysis of students' written production and its use in pedagogical decision-making. This paper presents the ongoing development of a Minimum Viable Product (MVP) aimed at supporting the qualified linguistic analysis of texts produced by deaf learners. Rather than functioning as a correction tool, the system is conceived as a pedagogically oriented analytical assistant, designed to identify patterns related to formal aspects of writing and their functional-communicative impact. The proposed model is structured around three analytical subdimensions (FORM_PROD, FORM_IMP, and FORM_FUN) and implemented through a prompt-based architecture using a Large Language Model (LLM). Preliminary results from a pilot study with five learner texts suggest the feasibility of distinguishing between formal instability and communicative effectiveness, generating interpretable and pedagogically relevant outputs. These findings point to the potential of AI to support language teaching in bilingual contexts involving deaf learners.

2012 ACM Subject Classification Applied computing → Education; Computing methodologies → Natural language processing

Keywords and phrases Deaf education, Second language writing, Linguistic analysis, Generative AI, Natural Language Processing

Digital Object Identifier 10.4230/OASICS.SLATE.2026.11

Category Short Paper

Funding São Paulo Research Foundation – FAPESP n.2022/05908-0

Teaching, Research and Extension Support Fund – FAEPEX/UNICAMP n. 3348/24.

1 Introduction

The teaching of writing as a second language (L2) to deaf students remains one of the critical dimensions of bilingual deaf education. In Brazil, despite significant legal advances, such as the establishment of parameters for bilingual deaf education (Decree No. 5.626/2005) and its institutionalization as an educational modality (Law No. 14.191/2021), a gap persists between public policies and pedagogical practices. In everyday school contexts, these advances have not been consistently translated into consolidated methodologies, appropriate teaching materials, or structured teacher education programs capable of addressing the specificities of writing development among deaf students.

At the core of this issue lies a linguistic and educational challenge: how to teach a written language to learners whose first language is visual-gestural-spatial in nature. In the Brazilian context, deaf students have Brazilian Sign Language (Libras) recognized as their first language (L1), while written Portuguese is introduced as L2 under conditions often marked by delayed exposure to language, limited access to linguistic input, and pedagogical



© Aryane Santos Nogueira and Ana Célia Ribeiro Bizigato Portes;
licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 11; pp. 11:1–11:10

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

practices insufficiently oriented toward bi/multilingualism. Consequently, the difficulties observed in these students' written productions are often interpreted from deficit-based perspectives, rather than being understood as manifestations of complex, bi/multilingual and multimodal linguistic repertoires [25, 26].

In this context, teachers tend to rely on generalized impressions of students' writing, without access to analytical tools capable of adequately and pedagogically identifying the linguistic patterns underlying these productions. This situation reveals a disconnection among three central dimensions of teaching practice: the analysis of students' productions, the assessment of their linguistic needs, and the planning of pedagogical interventions. Such disconnection compromises the effectiveness of teaching strategies and contributes to the maintenance of low expectations regarding deaf students' learning. Advancing the teaching of Portuguese as an L2, in this context, requires shifting the focus from generalized assessments to processes of linguistic analysis grounded in actual production data.

From this perspective, linguistic analysis [3], understood as a reflective and theoretically grounded practice focused on language in use, emerges as a promising methodological alternative. By shifting the focus from error correction to the identification of patterns related to textual organization, cohesion, enunciation, and argumentation, this approach enables a more qualified understanding of students' writing. However, its implementation in everyday school contexts is constrained by the concrete conditions of teaching work, particularly the limited available time for analyzing student productions and the lack of tools capable of systematically organizing and interpreting the linguistic data produced in the classroom.

Within this scenario, Generative Artificial Intelligence (GenAI) can be mobilized as a resource to support teaching practices, particularly with regard to the systematization and analysis of written productions. When designed in a critical and context-sensitive manner, AI-based tools can support the identification of linguistic patterns, contributing to more precise diagnoses of learning needs and to more informed pedagogical decision-making. Such technologies do not replace teachers' work; rather, they extend it by supporting analytical tasks while preserving the teacher's role in interpretation, mediation, and instructional design.

Based on these premises, this paper presents the development of an AI-integrated application, which aims to support a qualified linguistic analysis of written productions by deaf learners of written Brazilian Portuguese. Grounded in Corpus Linguistics approaches and in the principles of linguistic analysis in education, the proposed system seeks to contribute to overcoming the gap between diagnosis and pedagogical action by (i) identifying recurrent linguistic patterns in learners' writing and (ii) supporting teachers in the design of pedagogical strategies responsive to these patterns. By articulating principles from Applied Linguistics, language technologies, and Artificial Intelligence, this work engages with the fields of Human-Human Languages (HHL) and Human-Computer Languages (HCL), particularly in relation to the development of language technologies oriented toward supporting the analysis of written production and pedagogical decision-making in inclusive educational contexts.

2 Theoretical background

2.1 Deaf education and the teaching of writing

The teaching of writing to deaf students has been approached from different perspectives that articulate linguistic [37, 10, 8, 9], cognitive [22], pedagogical [7, 32, 34, 20], and technological dimensions [29, 35]. Overall, these studies converge on the understanding that writing development in this context is closely related to access to sign language [2, 30, 1], the

development of metalinguistic competencies [23, 39], and the adoption of pedagogical practices that are sensitive to the visual-gestural nature of communication used by deaf students [32, 13, 26]. In this sense, writing instruction tends to be more effective when grounded in bi/multilingual and multimodal approaches [12, 17, 25, 26], oriented toward language use and articulated with the integration of educational technologies.

Despite these advances, important gaps persist in the field, particularly with regard to teacher education and the consistency of pedagogical practices. Teachers frequently report not feeling prepared to teach writing to deaf students, especially in contexts marked by language deprivation [38, 36]. This is reflected in practices that remain insufficiently consistent, often centered on superficial revisions and with limited explicit instruction on aspects such as text genres and argumentation [37]. In addition, the integration of sign language and written language remains a recurring challenge [6], highlighting difficulties in implementing bi/multilingual approaches, both in defining the roles of languages and in developing coherent curricula [33].

From a linguistic perspective, structural differences between sign language and written language pose challenges, particularly with regard to morphosyntactic organization and textual cohesion [15]. At the same time, there is a lack of systematically evaluated and disseminated methods for teaching writing centered on deaf learners [14], as well as a scarcity of assessment instruments that take bi/multilingualism into account and avoid deficit-oriented biases [24]. In the technological domain, available educational resources are often incomplete, not intuitive, and difficult for teachers to implement [21]. At the same time, empirical evidence on the impact of emerging technologies, such as Artificial Intelligence and multimodal approaches, on the writing development of deaf students remains limited [19].

In the Brazilian context, these issues take on specific characteristics related to the historical trajectory of deaf education and the conditions for implementing bilingual education. Although legal frameworks have recognized Brazilian Sign Language (Libras) as the first language and established bilingual education as an educational modality, a significant gap remains in teacher education, particularly in preparing teachers to teach Portuguese as a second language to deaf students. Teacher education curricula tend to prioritize structural aspects of Libras at the expense of pedagogical approaches focused on writing instruction [27]. Furthermore, classroom practices often reproduce models oriented toward hearing students [10], overlooking the specificities of deaf bi/multilingualism and limiting the use of approaches based on the linguistic analysis of students' productions [28]. As a consequence, recurrent difficulties in writing, often described in generic terms, are not accompanied by systematic analyses that enable more precise and responsive pedagogical interventions and guidance. It is within this context that the proposal of qualified linguistic analysis of deaf learners' written production is situated, through an AI-integrated application conceived as a theoretical-methodological approach to support teaching practices in Portuguese as an L2 in bilingual contexts.

2.2 AI technologies in education

Recent advances in Artificial Intelligence (AI) have expanded the use of technologies in the teaching of reading and writing. Machine learning models, particularly those based on deep learning architectures, enable the identification of patterns in large volumes of linguistic data [5, 18]. In educational contexts, the growing adoption of generative AI tools has raised important questions about their impact on teaching and assessment practices, especially in areas centered on textual production [11, 16]. Rather than restricting their use, current discussions emphasize the need to integrate these technologies critically into pedagogical practices, with the aim of enhancing learning processes and rethinking assessment strategies.

In language education, such technologies have been used both to support writing and to assist teachers in text analysis and activity design [4]. However, their application in bilingual contexts involving deaf students remains limited, particularly with regard to addressing the linguistic specificities of this population. In this context, the use of AI to support the identification of linguistic patterns and the organization of data derived from students' written productions [31] emerges as a relevant possibility, provided it is guided by clear pedagogical objectives and closely aligned with teaching practice. This is the direction taken in this paper, which explores the use of AI to support the qualified linguistic analysis of deaf learners' writing and the design of pedagogical interventions based on these data.

3 Proposed approach: AI-supported linguistic analysis for deaf education

3.1 System design

As a feasibility-oriented pilot study, the prompt proposed for the FORM dimension was applied to five short texts (T1 to T5) written by deaf learners of Brazilian Portuguese as L2. The objective of this initial stage was not to evaluate the performance or validity of the system, but rather to investigate whether the proposed analytical framework could be consistently applied across multiple texts, generating interpretable, transparent, and pedagogically relevant outputs. Therefore, the results presented here should be understood as preliminary evidence of feasibility rather than as a validation of the analytical model.

The current stage of this work involves the development of a Minimum Viable Product (MVP) aimed at supporting qualified linguistic analysis of written productions by deaf learners of Brazilian Portuguese as a second language (L2). Rather than functioning as an automatic correction tool, the system has been conceived as a pedagogically oriented analytical assistant, with the potential to support teachers in identifying relevant linguistic patterns and translating these findings into instructional decisions.

The first development cycle prioritizes the FORM dimension (formal aspects of writing). This decision is based on three considerations: (i) formal features are more directly observable at the textual surface and can therefore be more consistently operationalized in prompt-based analyses; (ii) patterns such as orthographic instability, verb inflection, agreement, segmentation, punctuation, and basic syntactic organization can be systematically detected in early-stage prototypes; and (iii) these aspects may affect textual readability and school-based assessment processes, making them a relevant starting point for supporting teaching practices.

The FORM dimension is subdivided into three complementary analytical layers, allowing for a distinction between formal deviation and functional/communicative issues:

- **FORM_PROD**: productive patterns that support textual construction;
- **FORM_IMP**: recurrent unproductive patterns that hinder textual functioning;
- **FORM_FUN**: the actual functional-communicative impact of formal difficulties.

3.2 Technical environment and development workflow

The prototype is being developed in a Python environment, selected for its accessibility, low infrastructure cost, collaborative potential, and compatibility with Python-based Natural Language Processing workflows. The current version relies on the GPT-5.5 Large Language Model (LLM) accessed via API. This architecture enables rapid prototyping, prompt refinement, and iterative testing prior to a planned migration to a specialized Small Language Model (SLM).

3.3 Analytical workflow

The current workflow consists of submitting the learner text in .txt format to the model along with a structured prompt, followed by the analysis of the FORM_PROD, FORM_IMP, and FORM_FUN subdimensions. The system then generates structured output including scoring, textual evidence, diagnostic interpretation, and pedagogical suggestions. The results are subsequently stored in JSON for further use and analysis.

3.4 Prompt structure and scoring procedure

The current prototype relies on a structured prompt designed to operationalize the analytical framework proposed for the FORM dimension. Rather than requesting a general evaluation of learner writing, the prompt guides the model through a sequence of predefined analytical steps aligned with the pedagogical objectives of the project. The prompt is organized into three analytical subdimensions: FORM_PROD (productive formal resources), FORM_IMP (unproductive formal patterns), and FORM_FUN (functional-communicative impact). For each learner text, the model is instructed to: (i) identify productive linguistic resources that contribute to textual construction; (ii) identify recurrent formal difficulties that may affect textual functioning; and (iii) evaluate the extent to which these difficulties actually compromise communicative effectiveness. To increase transparency and interpretability, the prompt requires the generation of textual evidence supporting each analytical judgment. The output is structured into predefined sections, including general characterization of the text, numerical scores, textual evidence, pedagogical diagnosis, teaching priorities, and instructional recommendations. This design aims to support teachers not only in identifying linguistic patterns but also in translating analytical findings into pedagogical action. Scores are assigned on a five-point ordinal scale. For FORM_PROD, higher scores indicate a greater presence of productive resources that support textual construction. For FORM_FUN, higher scores indicate greater preservation of communicative effectiveness. In contrast, the logic of FORM_IMP is inverted: lower scores indicate a higher recurrence and impact of unproductive patterns, whereas higher scores indicate fewer relevant formal difficulties. The scoring procedure is descriptor-based rather than statistically derived. Each score corresponds to a predefined qualitative description associated with the observed linguistic patterns. Consequently, the generated scores should be interpreted as pedagogical indicators intended to support qualitative analysis rather than as psychometric measurements. At the current stage, the analyses are performed using a general-purpose Large Language Model accessed through an API. Model parameters were maintained at their default configuration during the pilot study. Because the system relies on generative AI, minor variations in textual explanations may occur across different executions. However, the analytical framework seeks to reduce variability by constraining the model through explicit instructions, predefined analytical categories, and structured output requirements.

3.5 Incremental strategy toward a specialized SLM

The current prompt-based version represents an exploratory stage within a broader strategy aimed at developing a specialized SLM, fine-tuned on manually annotated texts produced by deaf learners. It is expected that such specialized model will improve analytical consistency, reduce processing costs, and enhance control over the privacy of educational data.

The MVP currently under development is designed as a web-based application focused on the qualified linguistic analysis of learner texts. In this initial implementation cycle, priority is given to rapid prototyping, methodological validation, and iterative refinement of the analytical model, before any future scaling.

The application builds on previous experiences developed by the research team (FAPESP n.2022/05908-0) in the field of automated analysis of referential chains and anaphoric networks, in which Generative AI was integrated with structured prompts for the analysis of textual cohesion.

4 Preliminary results

As a pilot study, the prompt proposed for the FORM dimension was applied to five short texts (T1 to T5) written by deaf learners of Brazilian Portuguese as L2. The objective of this initial stage was not to establish the definitive performance of the system, but to verify whether the same analytical criteria could be consistently applied across multiple texts, producing interpretable and pedagogically relevant outputs.

The analyses were organized into three complementary subdimensions: FORM_PROD (productive patterns), FORM_IMP (unproductive patterns), and FORM_FUN (functional-communicative impact). This structure enabled the system to move beyond undifferentiated error detection by distinguishing linguistic resources, recurrent formal difficulties, and their actual effects on communication.

Scores ranged from 1 to 5 and were assigned based on previously defined descriptors for each subdimension. It is important to note that the logic of FORM_IMP is inverted: lower scores indicate a higher presence of recurrent unproductive patterns, whereas higher scores indicate a lower incidence of relevant difficulties.

■ **Table 1** Scoring Rubric for the FORM Dimension.

Score	FORM_PROD	FORM_IMP*	FORM_FUN
5	Systematic and consistent productive resources	Absence of relevant unproductive patterns	Communication preserved
4	Frequent productive resources with minor variation	Localized and occasional difficulties	Minor difficulties, comprehension maintained
3	Partial and irregular productive resources	Irregular difficulties with localized impact	Partial comprehension
2	Incipient productive resources	Frequent difficulties with relevant impact	Communication significantly impaired
1	Absence of productive resources	Systematic difficulties with high impact	Communication severely compromised
* In FORM_IMP, lower scores indicate greater recurrence or impact of difficulties.			

Across the five texts (T1 to T5 - Table 2), some degree of productive linguistic resources was identified in all cases, particularly lexical items and semantic cues that enabled the recovery of overall meaning. At the same time, recurrent unproductive patterns were consistently detected, especially those related to verbal morphology, agreement, orthographic instability, punctuation, and syntactic organization.

A relevant finding concerns the distinction between formal instability and communicative impact. Although FORM_IMP scores remained relatively similar across texts, FORM_FUN scores showed more variation. This suggests that frequent formal irregularities do not automatically imply communicative failure. Some texts with recurrent formal difficulties still preserved communicative intent and recoverable meaning.

This result is relevant, as traditional school-based assessments often reduce non-standard writing to a homogeneous category of deficit. In contrast, the proposed framework distinguishes textual resources, formal instability, and actual communicative effect. This

■ **Table 2** Pilot Results for the FORM Dimension.

Text	FORM_PROD	FORM_IMP	FORM_FUN	General Interpretation
T1	3	2	3	Partial productive resources; frequent formal difficulties; partial comprehension
T2	3	2	4	Partial productive resources; frequent formal difficulties; communication largely preserved
T3	3	2	4	Partial productive resources; recurrent difficulties; overall meaning preserved
T4	3	2	3	Partial productive resources; frequent formal difficulties; meaning recoverable with partial comprehension
T5	2	2	3	Incipient productive resources; frequent formal difficulties; meaning recoverable with partial comprehension

distinction is particularly relevant in bilingual deaf education contexts, in which texts may reflect processes of linguistic interaction (sign language-written language) rather than a lack of conceptual competence.

■ **Table 3** Pedagogical Priorities Generated by the Automated Analysis.

Text	Main Teaching Priorities	Suggested Pedagogical Focus
T1	Verb inflection; sentence organization; functional words	Guided rewriting; sentence expansion; use of connectors
T2	Number and subject-verb agreement; verbal regularity; orthographic revision	Contrastive revision; second version production
T3	Verb inflection; connectors; lexical revision	Rewriting routines; explicit relational markers
T4	Verb inflection; cohesion; agreement	Sentence reconstruction; context-oriented grammar
T5	Temporal organization; verb inflection; punctuation	Narrative sequencing; punctuation support activities

Another relevant result concerns the pedagogical usefulness of the generated reports (Table 3). Rather than merely listing formal issues, the system translated analytical findings into teaching priorities and classroom-oriented intervention strategies tailored to each text. This suggests that GenAI can support not only linguistic diagnosis but also responsive pedagogical decision-making.

The text excerpts (evidence T1 to T5 – Table 4) reinforce that the system does not rely solely on numerical scores. Instead, each analytical judgment is linked to textual evidence, enhancing interpretability, transparency, and pedagogical usefulness. This feature is particularly important in educational applications, where teachers require understandable justifications for analytical outputs.

■ **Table 4** Illustrative Textual Evidence from the FORM Analysis.

Text	Subdimension	Textual Evidence	Interpretation
T1	FORM_PROD	“Eu sonho muito viagem”	Lexical resources support the reconstruction of an intended life project
T1	FORM_IMP	“Eu quero ganha”	Instability in verb inflection affects formal adequacy
T1	FORM_FUN	“Melhor minha vida”	Overall meaning remains recoverable, though requiring inference
T2	FORM_IMP	“uma carro novo”	Recurrent instability in noun phrase construction
T3	FORM_FUN	“manter guardando o dinheiro=[dinheiro]”	Communicative intent is preserved despite formal instability
T4	FORM_IMP	“emprestado do dinheiro deixa-lá”	Syntactic organization and lexical choice reduce readability
T5	FORM_FUN	“mas perdeu infelizmente dinheiro”	Narrative progression is recoverable despite low formal stability

5 Conclusions – challenges and limitations

The results of the pilot study are promising; however, it is important to highlight the limitations of the current stage. The dataset is small and exploratory, and the outputs depend on prompt design, potentially varying across models or API configurations. Comparisons with expert analyses have not yet been conducted, although this is a planned step in the research. Therefore, the current findings should be interpreted as preliminary evidence of feasibility rather than as definitive validation.

References

- 1 T. E. Allen, A. Letteri, S. H. Choi, and D. Dang. Early visual language exposure and emergent literacy in preschool deaf children: Findings from a national longitudinal study. *American Annals of the Deaf*, 2014. doi:10.1353/aad.2014.0030.
- 2 T. E. Allen and D. A. Morere. Early visual language skills affect the trajectory of literacy gains over a three-year period of time for preschool aged deaf children who experience signing in the home. *PLoS ONE*, 2020. doi:10.1371/journal.pone.0229591.
- 3 M. A. Bezerra and M. A. Reinaldo. *Análise linguística: afinal a que se refere?* Editora Pipa Comunicação / Editora UFCG, Recife / Campina Grande, 2020.
- 4 J. Burstein, J. Sabatini, and J. Shore. Natural language processing for educational applications. In R. Mitkov, editor, *The Oxford Handbook of Computational Linguistics*. Oxford Academic, 2nd edition, 2014. doi:10.1093/oxfordhb/9780199573691.013.001.
- 5 K. R. Chowdhary. *Fundamentals of Artificial Intelligence*. Springer, 2020. doi:10.1007/978-81-322-3972-7.
- 6 H. M. Dostal and K. A. Wolbers. Developing language and writing skills of deaf and hard of hearing students: A simultaneous approach. *Literacy Research and Instruction*, 53(3):245–268, 2014. doi:10.1080/19388071.2014.907382.
- 7 H. M. Dostal, K. A. Wolbers, K. Spurgin, and L. Holcomb. Writing outcomes and expressive language use of deaf high school students. *The Journal of Deaf Studies and Deaf Education*, 31(1):70–84, 2026. doi:10.1093/jdsade/enaf050.
- 8 S. Fernandes. Avaliação em língua portuguesa para alunos surdos: algumas considerações. Cadernos PDE (Versão online), 2007. URL: <https://acervodigital.educacao.pr.gov.br/pages/view.php?ref=22722>.

- 9 S. Fernandes. Letramento na educação bilíngue para surdos: caminhos para a prática pedagógica. In M. C. L. Fernandes, M. J. Marçalo, and G. Micheletti, editors, *A língua portuguesa no mundo*. FFLCH, São Paulo, 2008. URL: <https://simelp.fflch.usp.br/sites/simelp.fflch.usp.br/files/inline-files/S2706.pdf>.
- 10 S. Fernandes, M. C. C. Pereira, and M. C. M. A. Ribeiro. *Português escrito para surdos [recurso eletrônico]*. Parábola, São Paulo, 2024.
- 11 L. Floridi. Ai as agency without intelligence: on chatgpt, large language models, and other generative models. *Philosophy & Technology*, 36(15), 2023. doi:10.1007/s13347-023-00621.
- 12 M. P. B. U. Francisco and P. P. Padilla. Using multimodal approach in teaching literacy to deaf college students. *Reading and Writing*, 2024. doi:10.1007/s11145-023-10440-4.
- 13 D. J. Griffin. American sign language and english bilingualism: educators' perspectives on a bicultural education. *International Journal of Bilingual Education and Bilingualism*, 2021. doi:10.1080/13670050.2018.1512552.
- 14 A. Guberman, T. Carmi, and R. Cohen-Brandeis. Devising writing instruction methods for deaf students in a teachers' professional learning community. *European Journal of Teacher Education*, pages 1–19, 2025. doi:10.1080/02619768.2025.2512951.
- 15 M. Gärdenfors and V. Johansson. Written products and writing processes in swedish deaf and hard of hearing children: an explorative study on the impact of linguistic background. *Frontiers in Psychology*, 14:1112263, 2023. doi:10.3389/fpsyg.2023.1112263.
- 16 N. K. Hayles. "don't ban ai from your writing classroom; require it!". *Poetics Today*, 45(2), 2024. doi:10.1215/03335372-11092870.
- 17 L. Holcomb. Writing development and translanguaging in signing bilingual deaf children of deaf parents. *Languages*, 2023. doi:10.3390/languages8010037.
- 18 D. Kaufman. *Desmistificando a inteligência artificial*. Autêntica, Belo Horizonte, 2022.
- 19 C. Kooli and R. Chakraoui. Ai-driven assistive technologies in inclusive education: benefits, challenges, and policy recommendations. *Sustainable Futures*, 10:101042, 2025. doi:10.1016/j.sftr.2025.101042.
- 20 A. C. B. Lodi, K. M. P. Harrison, and S. R. L. Campos. Letramento e surdez: um olhar sobre as particularidades dentro do contexto educacional. In A. C. B. Lodi, K. M. P. Harrison, S. R. L. Campos, and O. Teske, editors, *Letramento e minorias*, pages 35–46. Mediação, Porto Alegre, 2002.
- 21 A. Madhesh. Full exclusion during covid-19: Saudi deaf education is an example. *Heliyon*, 7(3):e06536, 2021. doi:10.1016/j.heliyon.2021.e06536.
- 22 M. Marschark. Interactions of language and cognition in deaf learners: from research to practice. *International Journal of Audiology*, 42(sup1):41–48, 2003. doi:10.3109/14992020309074623.
- 23 L. McQuarrie and M. Abbott. Bilingual deaf students' phonological awareness in asl and reading skills in english. *Sign Language Studies*, 2013. doi:10.1353/sls.2013.0028.
- 24 P. Ndibalema. Perspectives on barriers to learning opportunities among deaf children in tanzania. *International Journal of Educational Research Open*, 9:100468, 2025. doi:10.1016/j.ijedro.2025.100468.
- 25 A. S. Nogueira. O (in)esperado na descrição e compreensão da complexidade sociolinguística de práticas translingues de alunos surdos. *DELTA: Documentação de Estudos em Linguística Teórica e Aplicada*, 38(4):202259471, 2022. doi:10.1590/1678-460X202259471.
- 26 A. S. Nogueira. "É para escrever o português ou a libras?": nuances da translanguagem na educação linguística de surdos. *DELTA: Documentação de Estudos em Linguística Teórica e Aplicada*, 39(1):202359805, 2023. doi:10.1590/1678-460X202359805.
- 27 A. S. Nogueira and J. Cabello. O trabalho com narrativas audiovisuais no ensino de libras como l2 para ouvintes. *Revista Leitura*, 1(57), 2016. doi:10.28998/2317-9945.201657.320-347.
- 28 A. S. Nogueira, A. C. R. B. Portes, and J. B. Portes. Compilação e exploração de corpus textual de aprendizes surdos: projeto de aplicação web para apoio no ensino de português como segunda língua. *Trabalhos em Linguística Aplicada*, 64:e025041, 2025. doi:10.1590/01031813v64i120258679595.

- 29 P. Paim, S. S. Prietch, and J. A. Sánchez. Adapting codesign techniques for the construction of a learning environment of a written second language for the d/deaf. *Interacting with Computers*, 35(2):315–329, 2023. doi:10.1093/iwc/iwac023.
- 30 M. C. C. Pereira. O ensino de português como segunda língua para surdos: princípios teóricos e metodológicos. *Educar em Revista*, -(especial 2):143–157, 2014. doi:10.1590/0104-4060.37236.
- 31 C. M. de A. Pinho, M. A. Gaspar, and R. J. Sassi. Aplicação de técnicas de inteligência artificial para classificação de fuga em redações. *Educação em Revista*, 40:e39773, 2024.
- 32 K. Schonstrom and I. Holmström. Using l1 sign language to teach writing. In *The Routledge Handbook of Sign Language Pedagogy*. Routledge, 2019. doi:10.4324/9781315406824-6.
- 33 J. L. Singleton, S. Supalla, S. Litchfield, and S. Schley. From sign to word: Considering modality constraints in asl/english bilingual education. *Topics in Language Disorders*, 18(4):16–29, 1998. URL: https://journals.lww.com/topicsinlanguagedisorders/abstract/1998/08000/from_sign_to_word__considering_modality.4.aspx.
- 34 D. Steyer and C. A. Fronza. Reflexões sobre atividade de língua portuguesa escrita para alunos surdos. *Revista Entrelinhas*, 13(1), 2019. doi:10.4013/entr.2019.131.02.
- 35 M. S. Stinson. Current and future technologies in the education of deaf students. In *The Oxford Handbook of Deaf Studies, Language, and Education*. Oxford University Press, 2012. doi:10.1093/oxfordhb/9780195390032.013.0007.
- 36 K. Wolbers, H. Dostal, S. Graham, L. Branum-Martin, T. Allen, L. Holcomb, and R. Saulsbury. Writing knowledge, practices, efficacy, interests, attitudes, and beliefs of deaf education teachers: a randomized controlled trial. *Frontiers in Psychology*, 14:1214246, 2023. doi:10.3389/fpsyg.2023.1214246.
- 37 K. Wolbers, H. Dostal, L. Holcomb, and K. Spurgin. Writing instruction with grade-level/college-bound secondary deaf students. *The Journal of Deaf Studies and Deaf Education*, 29(2):145–157, 2024. doi:10.1093/deafed/enad062.
- 38 K. A. Wolbers. Using asl and print-based sign to build fluency and greater independence with written english among deaf students. *L1. Educational Studies in Language and Literature*, 10(1):99–125, 2009. doi:10.17239/L1ESLL-2010.10.01.05.
- 39 K. A. Wolbers, L. M. Bowers, H. M. Dostal, and S. C. Graham. Deaf writers' application of american sign language knowledge to english. *International Journal of Bilingual Education and Bilingualism*, 17(4):410–428, 2014. doi:10.1080/13670050.2013.816262.

OntoExpand: SPARQL-Based Ontology Expansion and Reasoning

Vitor Lelis N. Leite ✉ 

University of Minho, Braga, Portugal

José Carlos L. Ramalho ✉ 

University of Minho, Braga, Portugal

Abstract

The growing complexity and volume of OWL ontologies in the Semantic Web context present challenges for reasoning and knowledge expansion. Traditional ontology reasoners often face high computational costs, limiting their scalability and real-time applicability. Additionally there are other situations where reasoners crash and are not able to expand the ontology (for example, if your pattern includes property chains involving an irreflexive relation). This paper introduces OntoExpand, a new methodology for ontology expansion that uses SPARQL CONSTRUCT queries as an efficient alternative to conventional reasoning techniques. In OntoExpand we have created a mapping between OWL axioms and CONSTRUCT queries that are able to generate the new triples. These inferred triples are then reintegrated into the ontology using SPARQL INSERT queries, enabling incremental knowledge enrichment. This approach improves performance, reduces computational overhead, is pattern driven allowing a more granular expansion control and seamlessly integrates with SPARQL endpoints. OntoExpand is evaluated on multiple ontologies with varying sizes and complexities, demonstrating its scalability and practical utility in dynamic, ontology-driven environments. The results show that OntoExpand offers a viable and efficient mechanism for reasoning and expanding OWL ontologies, making it suitable for large-scale Semantic Web applications where traditional reasoning may be infeasible.

2012 ACM Subject Classification Software and its engineering → Search-based software engineering

Keywords and phrases OWL Ontologies, SPARQL, Reasoning, Knowledge Expansion, Inference

Digital Object Identifier 10.4230/OASICS.SLATE.2026.12

1 Context and Problem Statement

1.1 Challenges in Ontology Expansion

Ontologies play a fundamental role in knowledge representation by providing a structured, machine-processable model of concepts and their relationships within a domain [3]. Their importance is amplified in domains such as Artificial Intelligence, data integration, and especially the Semantic Web [1], where consistent interpretation of heterogeneous data is crucial.

However, as knowledge domains evolve and new data becomes available, maintaining ontologies becomes increasingly complex. A key challenge lies in the expansion of ontologies to accommodate new facts without compromising consistency. Traditionally, this task relies on external *reasoners* [2], which apply logical inference to generate new knowledge from existing axioms. Although effective, these systems introduce significant computational overhead, particularly in large-scale or real-time environments. Most reasoners process the ontology holistically, resulting in performance bottlenecks, high memory usage, and latency.

Beyond performance, current reasoners often struggle with certain constructs, such as the combination of property chains and irreflexive properties, which are essential for modeling complex relations but are not well supported in classical reasoning engines. This can limit the expressiveness and practical applicability of traditional reasoning in dynamic, real-world applications.



© Vitor Lelis N. Leite and José Carlos L. Ramalho;
licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 12; pp. 12:1–12:15

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1.2 Towards a Query-Based Reasoning Approach

To overcome these limitations, we propose **OntoExpand**, a reasoning and ontology expansion engine built upon SPARQL CONSTRUCT queries [6]. Unlike conventional reasoners, OntoExpand bypasses the need for external reasoning tools by translating ontological axioms into SPARQL constructs. This approach enables fine-grained, targeted inference directly over RDF data, improving both scalability and responsiveness.

SPARQL, being a W3C standard, provides a flexible and widely supported mechanism for querying RDF graphs. By leveraging CONSTRUCT queries, OntoExpand can infer new triples by applying rule-like transformations that mimic classical logical reasoning. This method is particularly well-suited for applications on the Semantic Web, where data is continuously evolving and systems must respond in real-time.

Furthermore, OntoExpand addresses specific reasoning gaps by explicitly supporting features such as property chains combined with irreflexivity constraints, cases that are problematic or unsupported in most existing reasoners. In doing so, it bridges the gap between expressiveness and performance, providing a reasoning model that is both practical and standards-compliant.

2 Methodology

2.1 Axiom-Driven Inference Using SPARQL

The core idea behind our approach is to replace traditional OWL reasoners with a query-based inference engine that leverages SPARQL CONSTRUCT queries to simulate logical reasoning over ontologies. To achieve this, we analyzed the OWL 2 specification [4] to identify key reasoning operations that could be directly mapped to SPARQL patterns. These reasoning operations are traditionally grounded in Description Logics and expressed through logical rules or axiomatic patterns.

Rather than relying on black-box reasoning tools, we manually translated selected OWL axioms into SPARQL query templates capable of deriving new RDF triples from existing assertions. Each SPARQL CONSTRUCT query acts as an application rule, injecting inferred knowledge back into the graph. This design allows us to model reasoning as an explicit, transparent, and tunable query process that can be selectively applied to a dataset.

2.2 Selected Inference Patterns

While OWL reasoning encompasses a wide variety of operations, in this paper we focus on two representative patterns that demonstrate the feasibility and benefits of our approach: **Instance Hierarchy Inference** and **Property Chain Inference**. The other patterns are only described.

2.2.1 Instance Hierarchy Inference

This pattern corresponds to the transitive propagation of class membership through subclass hierarchies.

$$\text{rdfs:subClassOf}(A, B) \wedge \text{rdf:type}(x, A) \rightarrow \text{rdf:type}(x, B) \quad (1)$$

This corresponds to the rule in Equation 1, inferring indirect class types for individuals by following subclass hierarchies.

■ **Listing 1** Instance Hierarchy query.

```

CONSTRUCT {
    ?indiv a ?parent .
}
WHERE {
    ?indiv a owl:NamedIndividual ;
        a ?class .
    ?class rdfs:subClassOf* ?parent .
    ?parent a owl:Class .

    FILTER NOT EXISTS {
        ?indiv a ?parent .
    }
}

```

Explanation.

- For each individual `indiv` typed with class `class`, the query finds all superclasses `parent` via `rdfs:subClassOf*`.
- Adds new type assertions `rdf:type(indiv, parent)` unless already present.

2.2.2 Property Chain Inference

OWL allows defining new properties as compositions of existing ones using `owl:propertyChainAxiom`.

$$R_1 \circ R_2 \rightarrow R_3 \Rightarrow (R_1(x, y) \wedge R_2(y, z) \rightarrow R_3(x, z)) \quad (2)$$

This type of inference is not always supported by conventional reasoners, especially when paired with constraints like irreflexivity. By explicitly encoding this inference pattern as SPARQL queries, our system enables the use of expressive reasoning constructs within the query model.

Property Chain Identification

In order to correspond to the condition of Equation 2, this query identifies the property chains equivalent to the original property.

■ **Listing 2** Property Chain SELECT query.

```

SELECT ?superProperty
    (GROUP_CONCAT(CONCAT("<", STR(?subProperty), ">"); SEPARATOR="/")
     AS ?propertyChain)
WHERE {
    ?superProperty owl:propertyChainAxiom ?chainList .
    ?chainList rdf:rest*/rdf:first ?subProperty .

    {
        ?subProperty a owl:ObjectProperty .
    }
    UNION {
        ?listNode rdf:first ?subProperty ;
            rdf:rest rdf:nil .
        ?subProperty a owl:DatatypeProperty .
    }
}
GROUP BY ?superProperty ?chainList

```

Explanation.

- Identifies each `owl:propertyChainAxiom` and its associated super-property.
- Builds the full chain of sub-properties using path concatenation.

Property Chain Inference

For each property chain discovered, the following inference is applied:

■ **Listing 3** Property Chain CONSTRUCT query.

```
CONSTRUCT {
    ?x <superProperty> ?z .
}
WHERE {
    ?x <property1>/<property2>/.../<propertyN> ?z .

    FILTER NOT EXISTS {
        <superProperty> a owl:IrreflexiveProperty .
        FILTER(?x = ?z)
    }
}
```

Explanation.

- Infers that `?x` is related to `?z` via the `<superProperty>` if there exists a matching chain of intermediate properties.
- The path expression `<property1>/<property2>/.../<propertyN>` represents the exact order defined in the chain.
- The `FILTER NOT EXISTS` clause ensures that the inference respects `owl:IrreflexiveProperty` semantics, i.e., it prevents assertions like `?x <superProperty> ?x` when such reflexivity is disallowed.
- This approach is particularly useful for modeling indirect or derived relationships (e.g., ancestry, delegation) while maintaining logical consistency with ontology constraints.

2.3 Mapping OWL to SPARQL

The methodology to map OWL patterns/axioms into SPARQL queries consists of three main steps:

1. **Axiom Pattern Identification:** We analyze OWL 2 constructs and define corresponding inference rules in logical form.
2. **SPARQL Template Design:** Each rule is translated into a reusable SPARQL CONSTRUCT query.
3. **Query Execution and Knowledge Expansion:** These queries are executed iteratively over an RDF graph, with inferred triples added dynamically.

By grounding inference in OWL semantics and expressing reasoning steps as explicit queries, this approach maintains alignment with Semantic Web standards while offering practical advantages in terms of scalability and flexibility.

It is important to note that the CONSTRUCT queries are used to generate inferred triples, but they do not modify the graph directly. To persist these results in the dataset, the inferred triples must be inserted using SPARQL INSERT queries [5]. This decouples inference from mutation, allowing better control over which inferences are materialized and when they are committed to the knowledge base.

3 Application example

The following example illustrates the methodology applied to a simple ontology.

3.1 Ontology Overview

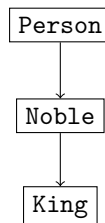
This ontology models a small fragment of the *House Targaryen* from the *Game of Thrones* universe. It was designed to demonstrate instance hierarchy and property chain inferences, two key reasoning patterns supported by OWL semantics.

3.1.1 Class Hierarchy

The ontology defines the following classes:

- **Person** – the most general class.
- **Noble** – a subclass of **Person**.
- **King** – a subclass of **Noble**.

The hierarchy can be visualized as:



3.1.2 Object Properties

The ontology defines several object properties:

- **hasParent** – links a child to a parent.
- **hasChild** – links a parent to a child.
- **hasSibling** – an `owl:IrreflexiveProperty` defined through a property chain:

$$\text{hasParent} \circ \text{hasChild} \Rightarrow \text{hasSibling}$$

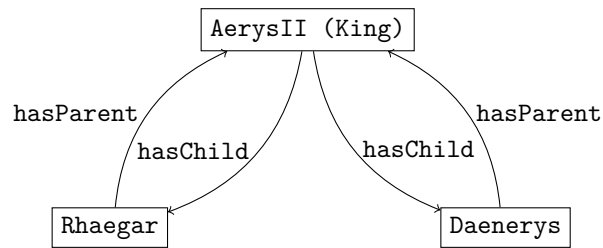
This chain means that two individuals are considered siblings if they share a parent.

3.1.3 Individuals and Relationships

Three key individuals are defined in the ontology:

- **Aerys II** is a **King**, and has two children: **Rhaegar** and **Daenerys**.
- **Rhaegar** is a **Person** and the child of **Aerys II**.
- **Daenerys** is a **Person** and also a child of **Aerys II**.

Their relationships can be represented as follows:



3.2 Inference Results

For the two selected patterns we present the new generated triples and the reasoning that led to them.

3.2.1 Instance Hierarchy Inference

In this ontology, the class hierarchy is structured as follows:

- King is a subclass of Noble
- Noble is a subclass of Person

The individual `AerysII` is explicitly typed as a `King`. Through subclass transitivity, it infers that `AerysII` is also a `Noble` and a `Person`.

These inferences are obtained using the query defined in Listing 1, which constructs new `rdf:type` statements for any superclass reachable via the subclass hierarchy.

- `AerysII rdf:type Noble`
- `AerysII rdf:type Person`

CONSTRUCT query results can then be added to the ontology by an INSERT query.

■ **Listing 4** INSERT for inferred types.

```

INSERT DATA {
  :AerysII a :Noble .
  :AerysII a :Person .
}
  
```

3.2.2 Property Chain Inference

The ontology defines the following property chain axiom:

`hasParent o hasChild ⇒ hasSibling`

This allows the reasoner to infer sibling relationships between individuals who share a common parent:

- `Rhaegar hasParent AerysII`
- `Daenerys hasParent AerysII`

And `AerysII` has both `Rhaegar` and `Daenerys` as children, we can infer:

- `Rhaegar hasSibling Daenerys`
- `Daenerys hasSibling Rhaegar`

These triples are generated using the construct pattern shown in Listing 3. The query includes a safeguard against asserting reflexive triples, considering the `owl:IrreflexiveProperty` type of `hasSibling`.

■ **Listing 5** INSERT for inferred siblings.

```
INSERT DATA {
  :Rhaegar :hasSibling :Daenerys .
  :Daenerys :hasSibling :Rhaegar .
}
```

4 Other Axioms

In this section, we present additional OWL reasoning patterns that can be translated into SPARQL-based inference rules using the same methodology described previously. For each axiom type, we provide a concise description of its semantic meaning, the logical rule that characterizes the inference, and the corresponding SPARQL query used to materialize the inferred triples.

4.1 Instance Checking

Instance checking determines whether an individual belongs to a particular class. This type of inference can be derived from *domain* property and *range* constraints.

Domain Inference

If a property has a declared domain, any individual appearing as the subject of that property can be inferred to belong to the domain class.

$$\text{rdfs:domain}(P, C) \wedge P(s, o) \rightarrow \text{rdf:type}(s, C) \quad (3)$$

The following query implements the rule in Equation 3.

■ **Listing 6** Domain Inference Query.

```
CONSTRUCT {
  ?indiv a ?class .
}
WHERE {
  ?indiv a owl:NamedIndividual .
  ?class a owl:Class .
  ?prop rdfs:domain ?class .
  ?indiv ?prop ?o .

  FILTER NOT EXISTS { ?indiv a ?class . }
}
```

Explanation.

- The query identifies which individuals appear as the subject of a property with a declared domain.
- It infers that the subject belongs to the domain class of that property.
- The `FILTER NOT EXISTS` clause prevents generating duplicate type assertions that are already present in the ontology.

Range Inference

Similarly, if a property has a declared range, any individual appearing as the object of that property can be inferred to belong to the range class.

$$\text{rdfs:range}(P, C) \wedge P(s, o) \rightarrow \text{rdf:type}(o, C) \quad (4)$$

The following query corresponds to the rule in Equation 4.

■ **Listing 7** Range Inference Query.

```
CONSTRUCT {
  ?indiv a ?class .
}
WHERE {
  ?indiv a owl:NamedIndividual .
  ?class a owl:Class .
  ?prop rdfs:range ?class .
  ?s ?prop ?indiv .

  FILTER NOT EXISTS { ?indiv a ?class . }
}
```

Explanation.

- The query identifies individuals that appear as the object of a property with a declared range.
- It infers that the object belongs to the range class of that property.
- The `FILTER NOT EXISTS` clause ensures that already existing type assertions are not duplicated.

4.2 Subclass and Subproperty Reasoning

Hierarchical relations between classes and properties allow the inference of more general knowledge from existing assertions. If a class is a subclass of another class, instances of the subclass can also be considered instances of the superclass. Similarly, if a property is defined as a subproperty of another property, triples using the subproperty imply corresponding triples using the superproperty.

$$\text{rdfs:subClassOf}(A, B) \wedge \text{rdfs:subClassOf}(B, C) \rightarrow \text{rdfs:subClassOf}(A, C) \quad (5)$$

$$\text{rdfs:subPropertyOf}(P, Q) \wedge P(x, y) \rightarrow Q(x, y) \quad (6)$$

Subclass Inference

The following query simulates the rule defined in Equation 5, retrieving inferred subclass relationships based on transitive subclass paths.

■ **Listing 8** Subclass Inference Query.

```

CONSTRUCT {
  ?sub rdfs:subClassOf ?super .
}
WHERE {
  ?sub a owl:Class .
  ?super a owl:Class .
  ?sub rdfs:subClassOf* ?super .

  FILTER(?sub != ?super)
  FILTER NOT EXISTS { ?sub rdfs:subClassOf ?super . }
}

```

Explanation.

- The query retrieves all subclass relations using the transitive path operator `rdfs:subClassOf*`.
- It identifies cases where a class is indirectly related to a superclass through intermediate classes.
- The `FILTER(?sub != ?super)` condition avoids reflexive subclass relations.
- The `FILTER NOT EXISTS` clause prevents generating subclass triples that already exist.

Subproperty Inference

The following query implements the rule in Equation 6, inferring superproperty triples based on subproperty relationships.

■ **Listing 9** Subproperty Inference Query.

```

CONSTRUCT {
  ?indiv ?superProp ?target .
}
WHERE {
  ?prop a owl:ObjectProperty ;
    rdfs:subPropertyOf+ ?superProp .
  ?indiv ?prop ?target .
  FILTER NOT EXISTS { ?indiv ?superProp ?target . }
}

```

Explanation.

- The query identifies properties that are part of a property hierarchy using `rdfs:subPropertyOf+`.
- For each triple `(?indiv ?prop ?target)`, it infers a new triple using the corresponding superproperty.
- The `FILTER NOT EXISTS` clause ensures that duplicate triples are not produced.

4.3 Inverse Property Inference

Inverse properties enable inference of new relationships by reversing existing triples according to inverse property declarations. If a property is defined as the inverse of another, then the existence of one relationship implies the corresponding reverse relationship.

$$\text{owl:inverseOf}(P, Q) \Rightarrow (P(x, y) \rightarrow Q(y, x)) \quad (7)$$

Additionally, the `owl:inverseOf` relation is defined as symmetric in OWL.

$$\text{owl:inverseOf}(P, Q) \rightarrow \text{owl:inverseOf}(Q, P) \quad (8)$$

Symmetric Inverse Declaration

Since ontologies may declare only one direction of an inverse property relation, the symmetric counterpart must be explicitly materialized before applying inverse inference. The following query enforces the rule in Equation 8.

■ **Listing 10** Inverse Property Symmetry Enforcement.

```
CONSTRUCT {
  ?inv owl:inverseOf ?prop .
}
WHERE {
  ?prop owl:inverseOf ?inv .
  FILTER NOT EXISTS { ?inv owl:inverseOf ?prop . }
}
```

Explanation.

- The query identifies properties connected through an `owl:inverseOf` relation.
- If the symmetric declaration does not exist, it constructs the reverse statement.
- This ensures that both directions of the inverse relation are explicitly represented in the ontology.

Inverse Inference

Once inverse relationships are fully declared, new triples can be inferred by reversing the direction of existing property assertions. The following query applies the rule in Equation 7.

■ **Listing 11** Inverse Property Inference Query.

```
CONSTRUCT {
  ?b ?inv ?a .
}
WHERE {
  ?prop a owl:ObjectProperty .
  ?inv a owl:ObjectProperty .
  ?prop owl:inverseOf ?inv .
  ?a ?prop ?b .
}
```

Explanation.

- The query identifies object properties that are declared as inverses.
- For each triple `?a ?prop ?b`, it infers the corresponding inverse triple `?b ?inv ?a`.
- This inference relies on the symmetric inverse relations established by the previous query.

4.4 Transitive Property Reasoning

Transitive properties allow the inference of indirect relationships through repeated applications of the same property. If a property is declared as transitive, the existence of two consecutive relationships implies a third relationship connecting the first and last individuals.

$$P \subseteq \text{owl:TransitiveProperty} \Rightarrow (P(x, y) \wedge P(y, z) \rightarrow P(x, z)) \quad (9)$$

Transitive Property Identification

The first step consists of identifying all properties declared as transitive in the ontology.

■ **Listing 12** Retrieve Transitive Properties.

```
SELECT ?prop WHERE {
  ?prop a owl:ObjectProperty, owl:TransitiveProperty .
}
```

Explanation.

- The query retrieves all object properties that are also declared as `owl:TransitiveProperty`.
- The resulting properties are then used to apply transitive inference over the dataset.

Transitive Inference

For each property obtained from the previous query, the following inference is applied.

■ **Listing 13** Deep Transitive Inference.

```
CONSTRUCT {
  ?x <{prop}> ?z .
}
WHERE {
  ?x <{prop}>+ ?z .
}
```

Explanation.

- The query searches for paths of one or more consecutive links using the transitive property.
- The `+` path operator enables traversal of chains of arbitrary length.
- For each discovered path between `?x` and `?z`, a direct triple is inferred.
- The placeholder `<prop>` is dynamically replaced with the property URI retrieved from Listing 12.

4.5 Symmetric Property Reasoning

Symmetric properties allow the inference of mirrored relationships between individuals. If a property is declared as symmetric, whenever a relationship exists in one direction, the corresponding reverse relationship can also be inferred.

$$P \subseteq \text{owl:SymmetricProperty} \Rightarrow (P(x, y) \rightarrow P(y, x)) \quad (10)$$

Symmetric Property Inference

The following query implements the rule in Equation 10.

■ **Listing 14** Symmetric Property Inference.

```
CONSTRUCT {
  ?subj ?prop ?target .
}
WHERE {
  ?prop a owl:ObjectProperty, owl:SymmetricProperty .
  ?target ?prop ?subj .
  FILTER NOT EXISTS { ?subj ?prop ?target . }
}
```

Explanation.

- The query identifies all object properties declared as `owl:SymmetricProperty`.
- For each triple (`?target ?prop ?subj`), it infers the corresponding symmetric triple (`?subj ?prop ?target`).
- The `FILTER NOT EXISTS` clause prevents generating duplicate triples that may already exist in the ontology.

4.6 Restrictions

OWL class restrictions allow the definition of classes based on constraints over properties. If an individual satisfies the conditions expressed by a restriction that is part of a class definition, it can be inferred to belong to that class. These restrictions are commonly expressed using constructs such as `owl:someValuesFrom`, `owl:allValuesFrom`, or `owl:hasValue`.

$$\text{rdf:type}(x, \exists P.C) \rightarrow \exists y. P(x, y) \wedge \text{rdf:type}(y, C) \quad (11)$$

Class Inference from Property Restrictions

The following query implements the rule in Equation 11.

■ **Listing 15** Class Inference from Restrictions.

```
CONSTRUCT {
  ?instance a ?inferClass .
}
WHERE {
  ?inferClass owl:equivalentClass/owl:intersectionOf ?restrictions .
  ?restrictions rdf:rest*/rdf:first ?part .

  OPTIONAL { ?part a owl:Restriction . }

  ?part owl:onProperty ?prop .
  ?part (owl:someValuesFrom | owl:allValuesFrom | owl:hasValue) ?val .

  { ?instance ?prop ?val .}
  UNION
  {
    ?instance ?prop ?anyValue .
    ?prop rdfs:range ?val .
  }
}
```

Explanation.

- The query identifies classes defined through `owl:equivalentClass` combined with `owl:intersectionOf`.
- Each element of the intersection is inspected to detect property restrictions.
- If an individual satisfies the required property constraint, the query infers that the individual belongs to the corresponding class.
- Restrictions using `owl:someValuesFrom`, `owl:allValuesFrom`, or `owl:hasValue` are supported.
- The query also allows inference when the property range implies the restricted class.

4.7 Equivalence & Subsumption Reasoning

Equivalence reasoning identifies when two classes can be considered logically equivalent based on the restrictions that define them. If two classes share the same property constraints, individuals satisfying those constraints can belong to either class, implying that the classes are semantically equivalent.

$$\forall x. (\text{rdf:type}(x, \exists P.A) \leftrightarrow \text{rdf:type}(x, \exists P.B)) \Rightarrow \text{owl:equivalentClass}(A, B) \quad (12)$$

Equivalence Class Detection

To correspond to Equation 12, this approach compares the property restrictions used in the definitions of different classes.

■ **Listing 16** Equivalence Class First Query.

```
SELECT ?class1
      (GROUP_CONCAT(CONCAT(STR(?p1), "->", STR(?v1)); SEPARATOR="|")
      AS ?rest1)
WHERE {
  ?class1
    owl:equivalentClass/owl:intersectionOf/rdf:rest*/rdf:first ?r1 .
  ?r1 a owl:Restriction ;
      owl:onProperty ?p1 ;
      (owl:someValuesFrom | owl:allValuesFrom | owl:hasValue) ?v1 .
}
GROUP BY ?class1
```

■ **Listing 17** Equivalence Class Second Query.

```
SELECT ?class2
      (GROUP_CONCAT(CONCAT(STR(?p2), "->", STR(?v2)); SEPARATOR="|")
      AS ?rest2)
WHERE {
  ?class2
    owl:equivalentClass/owl:intersectionOf/rdf:rest*/rdf:first ?r2 .
  ?r2 a owl:Restriction ;
      owl:onProperty ?p2 ;
      (owl:someValuesFrom | owl:allValuesFrom | owl:hasValue) ?v2 .
}
GROUP BY ?class2
```

■ **Listing 18** Inference of Equivalent Classes.

```

CONSTRUCT {
    ?class1 owl:equivalentClass ?class2 .
}
WHERE {
    {
        # First Query
    }

    {
        # Second Query
    }

    FILTER(?class1 != ?class2)
    FILTER(?rest1 = ?rest2)
}

```

Explanation.

- The first two queries extract the property restrictions that define each class.
- These restrictions are normalized into a signature string using `GROUP_CONCAT`.
- If two classes produce identical restriction signatures, they are inferred to be equivalent.
- The final query constructs an `owl:equivalentClass` relation between classes with matching definitions.

5 Conclusion

This paper presented a practical approach to simulating OWL reasoning operations using SPARQL queries, with a focus on instance hierarchy inference and property chain inference. By aligning OWL axioms with SPARQL `CONSTRUCT` queries, we demonstrated how implicit semantic relationships can be made explicit and persisted through `INSERT` operations.

Using a simplified ontology, we show how hierarchical class relations and property chain axioms can lead to meaningful inferences – such as superclass types and inferred sibling relationships – without relying on traditional OWL reasoners. The results highlight how SPARQL can be used to implement reasoning patterns in a transparent and explainable manner.

Although only two reasoning operations were presented in this paper, a broader set of OWL inference patterns were specified and described. Current work is focused on automating the generation of SPARQL queries for these operations with the goal of expanding coverage and improving scalability in more complex ontologies. This line of work contributes toward bridging the gap between declarative ontology semantics and practical query-driven inference over RDF data. We are also creating a web workbench where it is possible to select which patterns to apply and decide to expand or not the initial ontology.

References

- 1 Komal Dhulekar and Madhuri Devrankar. A review on semantic web. *International Journal of Engineering Technologies and Management Research*, 6:22–28, 2020. doi:10.29121/ijetmr.v6.i12.2019.470.
- 2 R. B. Mishra and Sandeep Kumar. Semantic web reasoners and languages. *Artificial Intelligence Review*, 35(4):339–368, 2011. doi:10.1007/s10462-010-9197-3.

- 3 Ontotext. What are ontologies?, 2025. URL: <https://www.ontotext.com/knowledgehub/fundamentals/what-are-ontologies/>.
- 4 W3C. OWL web ontology language reference. URL: <https://www.w3.org/TR/owl-ref/>.
- 5 W3C. SPARQL 1.1 update. URL: <https://www.w3.org/TR/sparql11-update/>.
- 6 W3C. SPARQL 1.1 query language, 2013. URL: <https://www.w3.org/TR/sparql11-query/>.

Deductive Databases and Logic Programming

Dietmar Seipel 

University of Würzburg, Germany

Salvador Abreu 

NOVA LINCIS, University of Évora, Portugal

Abstract

Logic programming is an excellent platform for implementing *declarative* rule-based systems in artificial intelligence (AI). It is also well-suited for mediating between declarative tools such as relational or *deductive databases*, semi-structured databases (XML), and semantic web databases (based on RDF or OWL). Often, domain-specific languages are used in expert systems based on user-defined infix operators.

For the evaluation of expert rule bases, the declarative *bottom-up* computation of Datalog known from deductive databases can be complemented with Prolog's *top-down* evaluation. We will review some related work on logic programming and deductive databases and describe some tools with examples for declarative logic programming in Prolog and Datalog for information systems in AI that have been implemented within the declarative toolkit **Declare**.

2012 ACM Subject Classification Theory of computation → Constraint and logic programming; Theory of computation → Logic and databases

Keywords and phrases Deductive Databases, Knowledge Bases, Logic Programming

Digital Object Identifier 10.4230/OASIS.SLATE.2026.13

Funding This work is supported in part by UID/04516/NOVA Laboratory for Computer Science and Informatics (NOVA LINCIS) with the financial support of FCT/IP.

1 Declarative Logic Programming

Declarative systems based on logic programming can combine declarative query languages such as SQL, SPARQL and *Datalog* with the processing of documents in established and widely used data formats, such as XML, RDF, and OWL. The logic programming language *Prolog* is an established and effective *mediator technology* between these hybrid data formats, since it allows for programming with meta-predicates and conveys expressiveness through nondeterminism built into the language. Also, Prolog is very well-suited for implementing *domain-specific languages* [6] using user-defined infix operators [23]. This feature promotes user-friendly applications, in which the language being coded in is embedded in the logic tools.

In what follows, we sketch the history of logic programming and databases, and then we briefly describe some declarative tools based on logic programming that we have implemented and that are mediated using Prolog.

1.1 Logic Programming and Databases

The field of logic and databases is an outgrowth of work in logic programming by Kowalski [10] and Lloyd [11] and in databases by Codd [3] and Ullman [26]. More powerful systems can be developed than by either approach alone, such as knowledge-based and nonmonotonic reasoning systems. Early work in relational databases emphasized implementation techniques and optimization methods using a *bottom-up* approach to query answering. This work led to efficient database systems of importance to commercial applications. Logic-based efforts



© Dietmar Seipel and Salvador Abreu;

licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 13; pp. 13:1–13:12

OpenAccess Series in Informatics



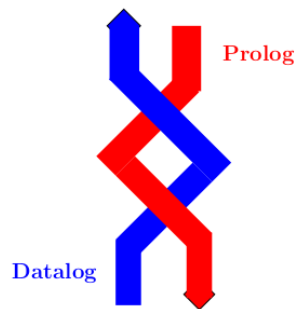
OASIS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

emphasized a *top-down* proof procedure and theoretical issues related to AI problems. The major contribution of Codd was to understand that the application of the relational calculus and the relational algebra to databases could yield a *declarative query language* that allows for *optimization techniques*. Those who approached databases through logic were influenced by work in theorem proving by J. Alan Robinson [19] who developed a uniform method to perform deduction in logic. The following development of the programming language Prolog by Colmerauer and his group [4] was very important for AI applications. Prolog is a top-down problem solver, and recursion is also permitted in logic programming while early relational systems did not permit recursion in queries or views. Early emphasis by the logic programming community was on syntax and semantics, only later commercially available systems could handle larger scale industrial problems.

In 1977, Gallaire, Nicolas and Minker organized a workshop on deductive databases [8]. Whereas logic programs permit function symbols in logical atoms, relational and deductive databases are generally function-free and deal almost exclusively with a finite set of constants in relations. The extended field of *disjunctive* logic programming and *disjunctive* deductive databases had its beginnings in the 80s [13, 12, 15]. The theoretical results extend the theory of logic programs as developed by Lloyd [11] and Apt [1]. The work in disjunctive logic programming has also been shown to be important for representing and implementing non-monotonic and abductive reasoning, and knowledge-based systems. There have been several *surveys* on deductive databases, starting with Gallaire, Minker and Nicolas [9]. Minker [14] provides historical material, results in complexity and early systems. Ramakrishnan and Ullman [18] focus on implementation techniques in Datalog, and they briefly describe a number of projects that have implemented systems. Minker, Seipel and Zaniolo [16] provide a history of the field of logic and databases from the late 1950s to the present.

1.2 Declare, DDBase, and Other Tools

The declarative logic programming system Declare with its deductive database component DDBASE provides a logic programming language Datalog* that combines the top-down reasoning of Prolog with the bottom-up reasoning of Datalog: In this paper, we will describe



■ **Figure 1** Prolog and Datalog for Declarative Logic Programming in DDBASE.

the tools DDBASE, FNQUERY, and DDQL of Declare with examples.

On the other hand, systems such as XSB Prolog [25] allows for top-down evaluation of logical rule systems in the programming style. XSB is an implementation of Prolog designed for knowledge-based systems, deductive databases, and AI applications in general. It extends Prolog with powerful features for efficient logical inference in situations not normally

handled by standard implementations. The key feature is *tabling* (*SLG resolution*): it stores intermediate results, which can be used to detect and avoid infinite loops, thereby ensuring termination for many recursive queries. Tabling has been the object of implementation in other Prolog systems, including Ciao [7], YAP [5], and SWI [28].

In contrast, standard Prolog may loop indefinitely and recompute the same results multiple times. Tabling is useful for recursive queries, graph traversal, and reasoning over complex rule systems. These are common in deductive databases, AI, and knowledge representation. Typical application areas in deductive databases are rule-based querying and recursive data relationships; in artificial intelligence, logical inference and knowledge representation can benefit; in the semantic Web, ontologies and rule-based reasoning; in program analysis, static analysis and security verification. A typical example in tabled Prolog is the well-known *ancestor* rule set. In summary, tabling is a powerful extension to Prolog, geared for efficient logical inference, and especially useful for recursive and knowledge-based applications over relations that include graphs with cycles.

Compared to tabled Prolog, the database-style *bottom-up* evaluation of DDBASE is easier to apprehend, as it separates normal Prolog from deductive database Datalog-style predicates explicitly. Moreover, *Declare* allows the embedding of Prolog calls into rule bodies, which are then evaluated as usual in logic programming languages. *Declare* can also make use of the more recently developed logic programming and non-monotonic reasoning (LPNMR) concept of *answer set programming* (ASP) on certain parts (strata) of the program and mediate between different types of hybrid knowledge bases.

The rest of this paper is organized as follows: Section 2 describes the tools DDBASE, FNQUERY, and DDQL of *Declare* in general, whereas Section 3 gives two more detailed examples for explaining the features of DDBASE. Finally, Section 4 closes with a summary and some remarks on possible future extensions.

2 Tools in Declare

For complex knowledge bases, specialized problem solvers might be embedded in *Declare*, e.g. *answer set programming* (ASP) solvers. For many practical applications, however, the expressivity and performance of *Declare* is already sufficient.

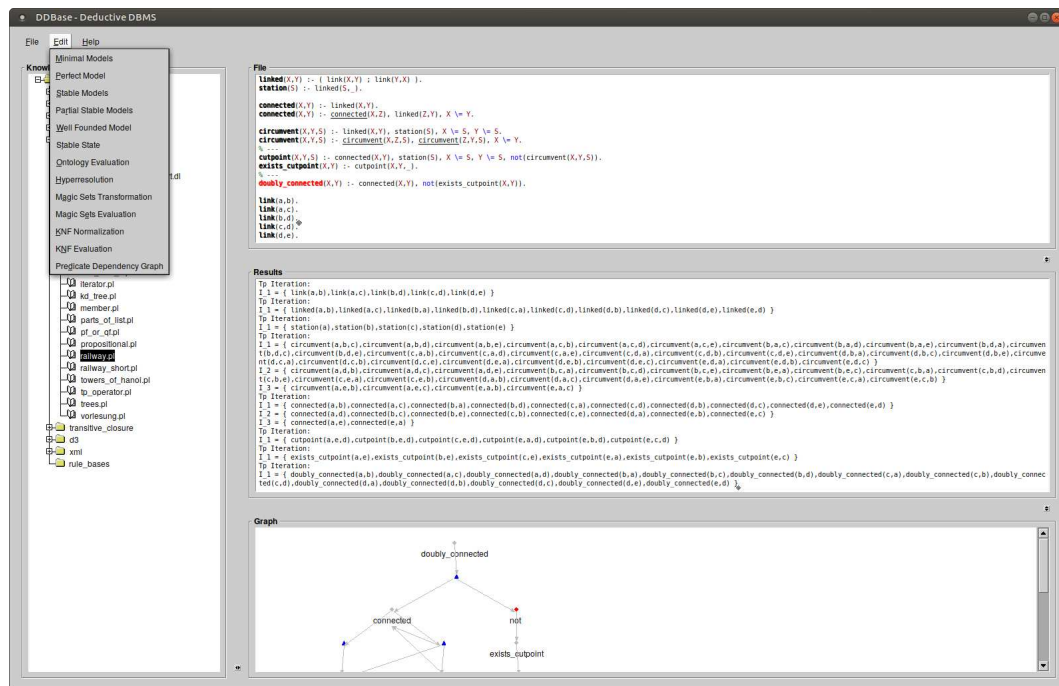
In the system *Declare*, the *bottom-up computation* of Datalog known from deductive databases is complemented with Prolog's *top-down evaluation*. Datalog is embedded and interleaved with Prolog, and Prolog goals can even be called from Datalog in the sense of built-in predicates, thereby gaining expressiveness, see Figure 1.

In many application domains, the bottom-up point of view is more natural and easier to understand for the users. Other approaches from the literature seek to make Prolog more declarative, e.g. by forbidding certain predicates with side effects.

2.1 The Rule Knowledge Base DDBase

Modern knowledge base systems frequently need to combine a collection of databases in different formats: e.g., relational databases, XML databases, rule bases, ontologies, etc. In the *deductive database system* DDBASE [21], we can manage these different formats of knowledge and reason about them – even the file systems on different computers could be part of the knowledge base. The Graphical User Interface of DDBASE is shown in Figure 2.

Often, it is necessary to handle different versions of a knowledge base; e.g., we might want to find out common parts or differences of two versions of a relational database.



■ Figure 2 The Graphical User Interface of DDBASE.

DDBASE uses the domain-specific extension Datalog* of the standard query language Datalog known from deductive databases. Syntactically, Datalog* allows for general Prolog rules with function symbols and meta-predicates (one of them is default negation), which must be *stratified*, i.e. no recursion is allowed through meta-predicates. The user is responsible for avoiding infinite loops.

In contrast to standard Prolog rules, the rules are evaluated bottom-up after a static program analysis. DDBASE uses abstractions of rule bases by predicate dependency and rule predicate graphs; also the proof trees of derived atoms can help to compare different versions of a rule base. A recent new contribution of DDBASE also allows for a *semi-stratified* evaluation of arbitrary default negation, that detects non-stratified parts by program analysis that are then evaluated using an ASP solver.

Moreover, it might be possible to have derivations joining rules with other formalisms of knowledge representation. Ontologies have shown their benefits in many applications of intelligent systems, and there have been many proposals for rule languages compatible with the semantic web stack, e.g., SWRL, the semantic web rule language. Recently, ontologies are used in hybrid systems for specifying the provenance of the different components.

DDBASE can compute the *perfect model* for a Datalog* program by repeated iterations on the strata, see Figure 2.

In the next section, we will give two detailed motivating examples for DDBASE, a longer one with stratified default negation on doubly connected networks, and a shorter one for grouping and counting the authors in a publication database with stratified aggregation.

2.2 XML Transformations, Queries, and Updates

With the tool FNQUERY [20], semi-structured (e.g. XML) data can be represented and queried nicely in a logic programming environment. It introduces a document object model for XML documents in Prolog, which is called *field notation*, and it simplifies dealing with

semi-structured data; instead of accessing a component of a Prolog fact by its argument position, one may do so non-positionally by its attribute name. One may also access nested components by a complex path or tree expression.

FNQUERY implements a library FNPATh of the domain-specific language XPATH from XML with predicates dealing with XML data in field notation. Practical case studies have shown how to extract and transform information from XML documents in a declarative style [20]. For some applications, the relevant information can be located automatically within the document, such that the extraction process becomes robust to certain changes of the document structure, over time.

As an example, consider an XML representation of a tennis match of the following form:

```
<match>
  <set id="1">
    <game id="1" service="A">
      <point id="1" top="B" winner="A">
        <hit id="1" hand="forehand" time="00:00:42" x="0.17" y="-12.07"/>
        <hit id="2" hand="backhand" time="00:00:44" x="-0.49" y="5.89"/>
        <hit id="3" hand="backhand" time="00:00:46" x="-3.92" y="-3.42"/>
        <hit id="4" hand="forehand" time="00:00:48" x="3.56" y="2.06"/>
      </point>
      <point id="2" ...> ... </point> ...
    </game> ...
  </set> ...
</match>
```

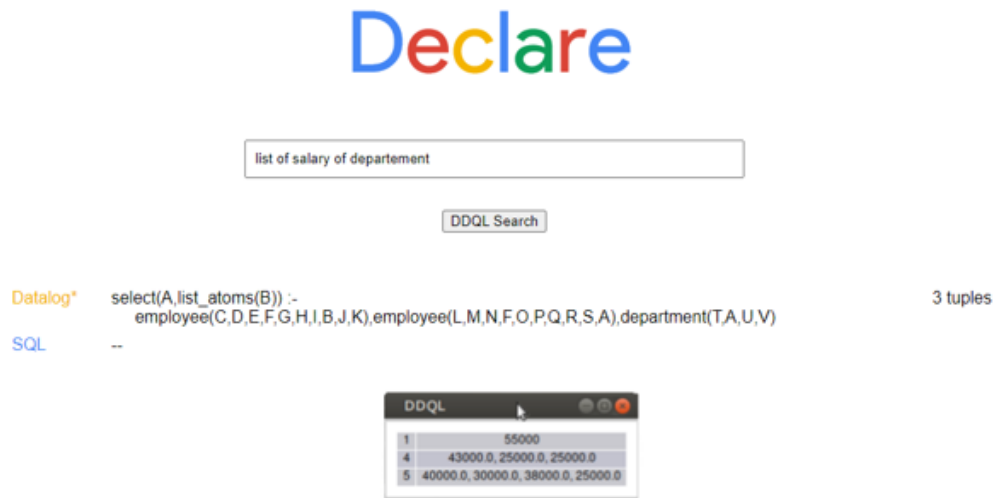
The following DDBASE query computes a list of tuples [S, G, P, L, D], where for each set S, game G, and point P, it computes L as the average of the points lengths and D as the average of the points distances; it uses path expressions in FNPATh to access the XML document Match:

```
average_point_distances(Match, Tuples) :-
  ddbase_aggregate( [S, G, P, avg(L), avg(D)],
    ( Point := Match/set::[@id=S]/
      game::[@id=G, @service=Service]/point,
      P := Match/player::[@id=Service]@name,
      point_to_length_and_distance(Point, L, D) ),
    Tuples ).
```

2.3 The Deductive Database Query Language DDQL

DDQL [24] proposes a natural, keyword-based query interface for knowledge bases – including relational or deductive databases – based on contextual background knowledge such as suitable *join conditions* or *synonyms*. Join conditions could be extracted from existing referential integrity (foreign key) constraints of the database schema; they could also be learned from other, previous database queries, if the database schema does not contain foreign key constraints.

Given a textual representation – a word list – of a query to a relational database, one may parse the list into a structured term. The intelligent and cooperative part of the approach is to hypothesize the semantics of the word list and to find suitable links between the concepts mentioned in the query using contextual knowledge, more precisely join conditions between the database tables.



■ **Figure 3** The Graphical User Interface of DDQL.

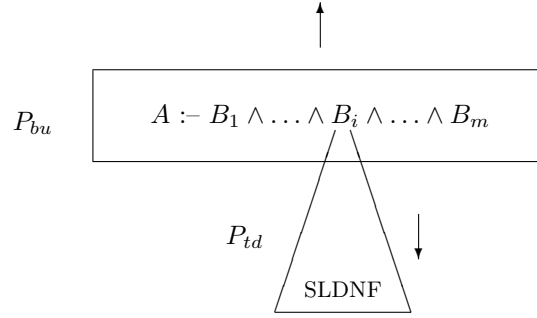
DDQL uses a knowledge-based parser based on an extension of *definite clause grammars* (DCG) that are interleaved with calls to the database schema to suitably annotate the tokens as table names, table attributes, attribute values or relationships linking tables. DDQL yields the possible queries in a special domain-specific rule language that extends Datalog, from which the user can choose one.

The deductive database query language DDQL has been described in detail with examples in another paper at SLATE 2023, cf. [22]. The example in Figure 3 shows the graphical user interface of DDQL for a relational database query asking for the list of salaries of department. This stimulates an aggregation query to group the salaries of the departments by listing all salaries per department. Since the `list` aggregation function is not allowed in SQL (the typical aggregation functions would be `sum`, `agg`, `min`, and `max`), this query can only be answered in DDBASE.

3 DDBase Examples

Deductive databases offer recursive rules, which can be seen as an extension of SQL views, where only limited forms of recursion are possible. In DDBASE, also stratified aggregation operations and default negation are possible; stratification means that the goal of the meta-operation (e.g. aggregation or default negation) has to be evaluated before the operation itself and cannot recursively depend on the meta-operation.

In deductive databases, we split a logic program P into subsets of rules (*strata*), such that a default negation in a stratum only refers to the same stratum or to lower strata. Then, the logic program can be evaluated stratum by stratum – beginning with the lowest stratum. This derives the so-called *perfect model* of the program. Such a partitioning into strata does not always exist, if there is default negation interleaved with recursion. Then, we need more elaborated evaluation methods and semantics such as *well-founded* or *stable models*.



■ **Figure 4** Forward Chaining with Prolog Built-Ins.

DDBASE allows for rules in Prolog syntax, which are – however – evaluated bottom-up. The program is split into a set P_{bu} of rules that are evaluated bottom-up and another set P_{td} of rules that are evaluated top-down. The body atoms B_i of a rule in P_{bu} are evaluated top-down with Prolog’s SLDNF resolution. If the body evaluates to true, then the head A of a rule is derived.

The consequence operator of DDBASE for forward chaining with Prolog built-ins evaluates a Prolog program P_{bu} , which is given as a list of rules and is not part of the Prolog database. It uses a helper Prolog program P_{td} that is stored separately.

3.1 Doubly Connected Network with Stratified Default Negation

The following recursive DATALOG^{not} program of [2] decides whether a network is doubly connected: $P_{dc} = \{ r_1, \dots, r_{10} \}$. For visualization, we use the subway network (Schnellbahn Netzplan) of Munich, cf. Figure 5. We can solve it and explain it more declaratively with the combination of bottom-up and top-down of DDBASE, cf. Figure 4, rather than it could be done in Prolog or XSB alone; in realistic extensions, there could be additional body atoms B_i to be solved in Prolog during a stratified bottom-up evaluation.

The first 5 rules compute stations and connected nodes X and Y from a base relation *link*, which is first made symmetric in the form of *linked* and for which we also define its transitive closure, expressed as *connected*, as follows:

$$\begin{aligned}
 r_1 &= \text{linked}(X, Y) :- \text{link}(X, Y), \\
 r_2 &= \text{linked}(X, Y) :- \text{link}(Y, X), \\
 r_3 &= \text{station}(X) :- \text{linked}(X, Y), \\
 r_4 &= \text{connected}(X, Y) :- \text{linked}(X, Y), \\
 r_5 &= \text{connected}(X, Y) :- \text{linked}(X, Z) \wedge \text{connected}(Z, Y) \wedge X \neq Y.
 \end{aligned}$$

The following rules compute triples X, Y, S , such that there exists a connection from X to Y that circumvents S (rules r_6 and r_7):

$$\begin{aligned}
 r_6 &= \text{circumvent}(X, Y, S) :- \\
 &\quad \text{linked}(X, Y) \wedge \text{station}(S) \wedge X \neq S \wedge Y \neq S, \\
 r_7 &= \text{circumvent}(X, Y, S) :- \\
 &\quad \text{circumvent}(X, Z, S) \wedge \text{circumvent}(Z, Y, S) \wedge X \neq Y.
 \end{aligned}$$



Figure 5 Subway Network of Munich.

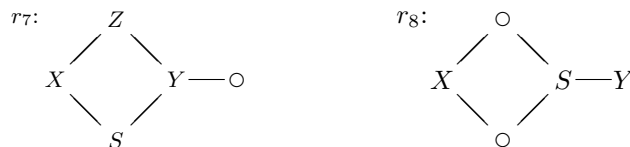


Figure 6 Circumvent and Cutpoint.

S is a cutpoint between X and Y , if X and Y are connected and S cannot be circumvented on this connection (rule r_8), see Figure 6:

$$\begin{aligned} r_8 = \text{cutpoint}(X, Y, S) :- \\ & \text{connected}(X, Y) \wedge \text{station}(S) \wedge \\ & X \neq S \wedge Y \neq S \wedge \\ & \text{not circumvent}(X, Y, S). \end{aligned}$$

According to rule r_8 , a station S is a cutpoint, if X and Y are connected, S is different from X and Y (no matter where S occurs on a path from X to Y), and S cannot be circumvented. In Figure 6, on the left, S can be circumvented, since there exists another path through Z ; on the right, S cannot be circumvented, since the only path from X to Y has to pass S .

Two nodes X and Y are doubly connected, if they are connected and there exists not cutpoint between X and Y (rules r_9 and r_{10}):

$$\begin{aligned} r_9 = \text{exists\_cutpoint}(X, Y) :- \\ & \text{cutpoint}(X, Y, S), \\ r_{10} = \text{doubly\_connected}(X, Y) :- \\ & \text{connected}(X, Y) \wedge X \neq Y \wedge \\ & \text{not exists\_cutpoint}(X, Y). \end{aligned}$$

In summary, the DATALOG^{not} program $P_{dc} = \{r_1, \dots, r_{10}\}$ has, e.g., the following stratification

$$\begin{aligned} \Pi_1 &= \{ \text{link}, \text{linked}, \text{station}, \text{connected}, \text{circumvent} \}, \\ \Pi_2 &= \{ \text{cutpoint}, \text{exists\_cutpoint} \}, \\ \Pi_3 &= \{ \text{doubly\_connected} \}, \end{aligned}$$

with the corresponding strata $P_1 = \{r_1, \dots, r_7\}$, $P_2 = \{r_8, r_9\}$, $P_3 = \{r_{10}\}$. The dependency graph $G_{P_{dc}}^d$ is shown in Figure 7. Besides the marked stratification, there exist others.

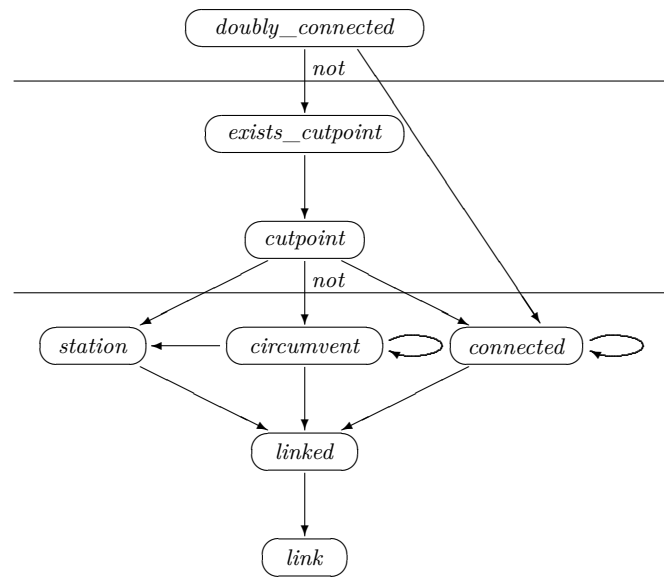
In DDBASE, this program P_{dc} can be evaluated using a stratified bottom-up fixpoint computation. Many intermediate facts are computed during the evaluation, but this poses no problem to the DDBASE system. In practice, the facts for `link/2` and `station/1` were derived by the following Prolog helper rules

```
link(X, Y) :-
    line(_, Stations), append(_, [X, Y|_], Stations).
station(S) :-
    line(_, Stations), member(S, Stations).
```

and facts for `line/2` of the form

```
line(s1, [
    freising, moosach, laim, donnersbergerbruecke, ostbahnhof,
    giesing, neuperlachsued, kreuzstrasse ]).
...
```

Together with P_{dc} , these rules form a Datalog* program P_{bu} . The helper program is formed by the standard Prolog rules for `append/3` and `member/2`. We got good performance on stratified evaluation, taking less than 17 seconds as more than 60.000 facts are generated and processed, running on a single-threaded Intel Core i7-8565U CPU.



■ **Figure 7** Dependency Graph with Stratification in 3 Strata.

3.2 Count Authors with Aggregation

Given a base predicate about publications, the following DDBASE code can compute coauthor counts and lists:

```
co_author(Author_1, Author_2) :-
    paper(Author_1, Paper),
    paper(Author_2, Paper),
    Author_1 \= Author_2.

co_authors(Author, Count, Co_Authors) :-
    [Author, Count, Co_Authors] <==
        ddbase_aggregate( [A, count(B), list(B)],
            co_author(A, B) ).
```

A call `Xs <== ddbase_aggregate(Template, Goal)` computes all answers `Template` to the goal `Goal`, and successively unifies the list `Xs` with these answers. Thus, lists can be generated by this construction, which is not possible by aggregation in SQL, and the aggregation functions can be self-defined.

For an example database on publications with binary facts about `paper`, transitive coauthors can already be computed in standard deductive databases. In DDBASE, the coauthor distance, Erdős number or cliques can be computed by built-in calls to Prolog or other external languages. If the base facts are given in domain-specific terms for XML, then FNQUERY can handle the transformation from this XML to Datalog.

In DDBASE, embedded calls to answer set programming (ASP solvers) can be used to evaluate non-stratified default negation – which can be detected by program analysis. In DDQL, natural queries in the form of Google-like buzzword lists can be evaluated. We are presently working on an extension with automated speech recognition (ASR) and contextual logic programming.

4 Conclusions and Future Directions

The deductive database component DDBASE of the logic programming toolkit Declare helps to effectively build applications of intelligent information systems in a declarative way. Datalog* can embed Prolog calls into the standard bottom-up evaluation of Datalog. Like in embedded SQL, Datalog can now be embedded into the programming language Prolog, and Prolog built-ins can be embedded into Datalog.

Relational databases and XML documents can be accessed from Datalog* by ODBC and FNQUERY, respectively. We have not shown ODBC here, since it is a standard feature of programming languages such as Prolog. The query language DDQL is another tool of Declare, which has been described in detail in another paper.

It is also our belief that an integration of the program-structuring abilities brought by contexts, especially along the lines of temporal contextual logic programming [17], can be leveraged to effectively and naturally build further *mediator programming* features, in which multiple versions of ontologies may coexist and be queried in a unified way. We are working on an integration of TCxLP into DDBASE and thereby provide a scalable and efficient means for keeping control over large-scale web or database applications. Such a system would inherit the underlying components' properties of verifiability and provability.

References

- 1 Krzysztof R. Apt. Logic Programming. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, pages 493–574. Elsevier and MIT Press, 1990. doi:10.1016/b978-0-444-88074-1.50015-9.
- 2 Stefano Ceri, Georg Gottlob, and Letizia Tanca. *Logic Programming and Databases*. Surveys in computer science. Springer, 1990. URL: <https://www.worldcat.org/oclc/20595273>.
- 3 E. F. Codd. A Relational Model of Data for Large Shared Data Banks. *Commun. ACM*, 13(6):377–387, 1970. doi:10.1145/362384.362685.
- 4 Alain Colmerauer, Henry Kanoui, R. Pasero, and P. Roussel. Un système de communication homme-machine en français. Technical report, Groupe de Intelligence Artificielle Université de Aix-Marseille II, 1973.
- 5 Vítor Santos Costa, Ricardo Rocha, and Luís Damas. The YAP Prolog system. *Theory Pract. Log. Program.*, 12(1-2):5–34, 2012. doi:10.1017/S1471068411000512.
- 6 M. Fowler. *Domain-Specific Languages*. Addison-Wesley, 2011.
- 7 Manuel V. Hermenegildo, Francisco Bueno, Manuel Carro, Pedro López-García, Edison Mera, José F. Morales, and Germán Puebla. An Overview of Ciao and its Design Philosophy. *Theory Pract. Log. Program.*, 12(1-2):219–252, 2012. doi:10.1017/S1471068411000457.
- 8 Herve Gallaire and Jack Minker, editor. *Logic and Data Bases*. Plenum Press, 1978.
- 9 Hervé Gallaire and Jack Minker and Jean-Marie Nicolas. Logic and Databases: A Deductive Approach. *ACM Comput. Surv.*, 16(2):153–185, 1984. doi:10.1145/356924.356929.
- 10 Robert A. Kowalski. Predicate Logic as Programming Language. In Jack L. Rosenfeld, editor, *Information Processing, Proceedings of the 6th IFIP Congress 1974, Stockholm, Sweden, August 5-10, 1974*, pages 569–574. North-Holland, 1974.
- 11 John W. Lloyd. *Foundations of Logic Programming, 2nd Edition*. Springer, 1987. doi:10.1007/978-3-642-83189-8.
- 12 Jorge Lobo, Jack Minker, and Arcot Rajasekar. *Foundations of Disjunctive Logic Programming*. Logic Programming. MIT Press, 1992.
- 13 Jack Minker, editor. *Foundations of Deductive Databases and Logic Programming*. Morgan Kaufmann, 1988. doi:10.1016/c2013-0-07699-2.
- 14 Jack Minker. Logic and Databases: A 20 Year Retrospective. In Dino Pedreschi and Carlo Zaniolo, editors, *Logic in Databases, International Workshop LID'96, San Miniato, Italy, July 1-2, 1996, Proceedings*, volume 1154 of *Lecture Notes in Computer Science*, pages 3–57. Springer, 1996. doi:10.1007/BFb0031734.

- 15 Jack Minker and Dietmar Seipel. Disjunctive Logic Programming: A Survey and Assessment. In Antonis C. Kakas and Fariba Sadri, editors, *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part I*, volume 2407 of *Lecture Notes in Computer Science*, pages 472–511. Springer, 2002. doi:10.1007/3-540-45628-7_18.
- 16 Jack Minker, Dietmar Seipel, and Carlo Zaniolo. Logic and Databases: A History of Deductive Databases. In Jörg H. Siekmann, editor, *Computational Logic*, volume 9 of *Handbook of the History of Logic*, pages 571–627. Elsevier, 2014. doi:10.1016/B978-0-444-51624-4.50013-7.
- 17 Vítor Nogueira and Salvador Abreu. Temporal Contextual Logic Programming. *Electron. Notes Theor. Comput. Sci.*, 177:219–233, 2007. doi:10.1016/j.entcs.2007.01.025.
- 18 Raghu Ramakrishnan and Jeffrey D. Ullman. A Survey of Deductive Database Systems. *J. Log. Program.*, 23(2):125–149, 1995. doi:10.1016/0743-1066(94)00039-9.
- 19 John Alan Robinson. A Machine-Oriented Logic Based on the Resolution Principle. *J. ACM*, 12(1):23–41, 1965. doi:10.1145/321250.321253.
- 20 Dietmar Seipel. Processing XML-Documents in Prolog. In *Proc. 17th Workshop on Logic Programming (WLP 2002)*, 2002.
- 21 Dietmar Seipel. Knowledge Engineering for Hybrid Deductive Databases. In *Proceedings 29th and 30th Workshops on (Constraint) Logic Programming and 24th International Workshop on Functional and (Constraint) Logic Programming, WLP 2015 / WLP 2016 / WFLP 2016*, volume 234 of *EPTCS*, pages 1–12, 2017. doi:10.4204/EPTCS.234.1.
- 22 Dietmar Seipel, Benjamin Förster, Magnus Liebl, Marcel Waleska, and Salvador Abreu. Querying Relational Databases with Speech-Recognition Driven by Contextual Knowledge. In Alberto Simões, Mario Marcelo Berón, and Filipe Portela, editors, *12th Symposium on Languages, Applications and Technologies, SLATE 2023, Vila do Conde, Portugal, June 26-28, 2023*, OASICS, pages 6:1–6:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/OASICS.SLATE.2023.6.
- 23 Dietmar Seipel, Falco Nogatz, and Salvador Abreu. Domain-specific Languages in Prolog for Declarative Expert Knowledge in Rules and Ontologies. *Comput. Lang. Syst. Struct.*, 51:102–117, 2018. doi:10.1016/j.cl.2017.06.006.
- 24 Dietmar Seipel, Daniel Weidner, and Salvador Abreu. Intelligent Query Answering with Contextual Knowledge for Relational Databases. In *10th Symposium on Languages, Applications and Technologies, SLATE*, volume 94 of *OASICS*, pages 16:1–16:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/OASICS.SLATE.2021.16.
- 25 Terrance Swift and David Scott Warren. XSB: Extending Prolog with Tabled Logic Programming. *CoRR*, abs/1012.5123, 2010. arXiv:1012.5123.
- 26 Jeffrey D. Ullman. *Principles of Database and Knowledge-Base Systems, Volume II*. Computer Science Press, 1989.
- 27 Jeffrey D. Ullman. The Theory of Deductive Database Systems. In *Intellectual Leverage: Thirty-Fifth IEEE Computer Society International Conference, Compcon Spring '90, San Francisco, California, USA, February 26 – March 2, 1992, Digest of Papers*, pages 496–502. IEEE Computer Society, 1990. doi:10.1109/CMPCON.1990.63730.
- 28 Jan Wielemaker, Tom Schrijvers, Markus Triska, and Torbjörn Lager. SWI-Prolog. *Theory Pract. Log. Program.*, 12(1-2):67–96, 2012. doi:10.1017/S1471068411000494.

An OWL/RDF Ontology for Declarative User Interface Generation

Daniel Theisges da Costa ✉

Research Centre in Digitalization and Intelligent Robotics (CeDRI),
Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC),
Instituto Politécnico de Bragança, Portugal

Júlio Costa ✉ 

Research Centre in Digitalization and Intelligent Robotics (CeDRI),
Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC),
Instituto Politécnico de Bragança, Portugal

Paulo Jorge Teixeira Matos ✉ 

Research Centre in Digitalization and Intelligent Robotics (CeDRI),
Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC),
Instituto Politécnico de Bragança, Portugal

Ramon Gallinad Corti ✉ 

Dept. de Telecomunicació i Enginyeria de Sistemes, Escola d'Enginyeria,
Universitat Autònoma de Barcelona, Spain

Ramon Vilanova Arbos ✉ 

Dept. de Telecomunicació i Enginyeria de Sistemes, Escola d'Enginyeria,
Universitat Autònoma de Barcelona, Spain

Maria João Varanda Pereira ✉ 

Research Centre in Digitalization and Intelligent Robotics (CeDRI),
Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC),
Instituto Politécnico de Bragança, Portugal

Abstract

Maintaining user interfaces for data-intensive applications is costly because interface code is tightly coupled to the underlying domain model: any change to an operation or data schema forces manual updates across multiple layers of the software stack. We present *OntologyToUI*, a framework that treats ontologies written in OWL (Web Ontology Language)/RDF (Resource Description Framework) as a Domain-Specific Language (DSL) for user-interface specification. Five modular base ontologies (schema-level definitions of abstract classes and properties, with no domain-specific instances) define the metamodel, covering communication technologies, external code references, data metadata, business operations, and User Interface (UI) element types. Domain-specific *view ontologies*, written in Turtle, instantiate the *OntologyToUI-UserInterface* ontology to declare UI elements. A generic React-based rendering engine interprets view ontologies at runtime through SPARQL (SPARQL Protocol and RDF Query Language) queries and a configuration-driven dispatch table, mapping element types to components without any domain-specific logic. We validate the framework through two case studies developed in the scope of the Smart_LEM project [11]: a Local Electricity Market management platform, where eleven views were created with zero application-code changes; and a real-time Internet of Things (IoT) dashboard that introduces a new Message Queuing Telemetry Transport (MQTT)-subscribed element type, demonstrating the framework's extensibility.

2012 ACM Subject Classification Software and its engineering → Domain specific languages; Software and its engineering → Model-driven software engineering; Information systems → Ontologies

Keywords and phrases OWL, RDF, domain-specific language, user interface generation, model-driven engineering, React, SPARQL, MQTT

Digital Object Identifier 10.4230/OASICS.SLATE.2026.14

Supplementary Material *Software (NPM Package source code)*: <https://github.com/NewDaniC/OntologyToUI> [4]; archived at `swb:1:dir:840123e5e0c6e231e56bad1d41fe75ba1890f803`



© Daniel Theisges da Costa, Júlio Costa, Paulo Jorge Teixeira Matos, Ramon Gallinad Corti, Ramon Vilanova Arbos, and Maria João Varanda Pereira;
licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 14; pp. 14:1–14:11

OpenAccess Series in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Funding This work was supported by FCT - Fundação para a Ciência e Tecnologia, I.P. by projects: CeDRI, UID/05757/2025 (DOI: <https://doi.org/10.54499/UID/05757/2025>) and UID/-PRR/05757/2025 (DOI: <https://doi.org/10.54499/UID/PRR/05757/2025>); SusTEC, LA/P/0007/2020 (DOI: <https://doi.org/10.54499/LA/P/0007/2020>) and funded by CETP, the Clean Energy Transition Partnership under the 2022 CETP joint call for research proposals, co-funded by the European Commission (GAN° 101069750). Specifically, the financial assistance of the FCT grant CETP/0003/2022 (DOI <https://doi.org/10.54499/CETP/0003/2022>) towards this research is hereby acknowledged.

1 Introduction

In this section, the context, motivation and main goals of this work are presented, along with an overview of the article structure.

Modern software applications expose their domain through two tightly coupled artefacts: a backend Application Programming Interface (API) and a frontend user interface. Any evolution of the domain (a new operation, an additional parameter, or a changed response schema) requires coordinated updates in both artefacts. In domains such as energy management or IoT monitoring, where operational rules and data models evolve frequently, this coupling becomes a bottleneck that consumes developer resources and introduces the risk of inconsistency across the software stack [2].

Model-Driven Engineering (MDE) addresses this class of problem by raising the primary development artefact from code to models, from which executable software can be generated automatically [2]. Applied to user interfaces, Model-Based User Interface Development (MBUID) proposes that interfaces be specified at an abstract level and derived through systematic transformations [3]. However, existing MBUID approaches rely on proprietary modeling languages that require specialized tooling and do not exploit the formal semantics, standardized query mechanisms, and reasoning capabilities offered by Semantic Web ontologies.

We propose *OntologyToUI*: a framework in which ontologies written in OWL/RDF serve as the single source of truth for both API generation and UI specification. *OntologyToUI* is developed in the scope of Smart_LEM [11], a business model-oriented platform for Local Electricity Markets; the complete architecture of the system, spanning API generation and UI specification, is illustrated in Figure 1. From a DSL perspective [7], the base ontologies constitute the *grammar*, each view ontology is a *program* written in that language, and a generic rendering engine is the *interpreter* that executes it. The key design goal is the *Generic Rendering Principle*: the engine must accept any ontology conforming to the established schema and render only what is declared therein, without hardcoding element lists or making domain assumptions.

OntologyToUI has been designed with three objectives:

- **Declarative UI specification:** allow domain experts to author view interfaces in Turtle without programming knowledge, by instantiating a controlled vocabulary of UI element types.
- **Generic interpretation:** implement a rendering engine that processes any conformant view ontology dynamically, driven entirely by SPARQL queries and a dispatch table, with no hardcoded domain logic.
- **Extensibility:** support new UI element types by adding one entry to a registry and one React component, without modifying the rendering pipeline or any existing component.

This article is structured as follows. Section 2 reviews the state of the art in model-driven UI development and ontologies. Section 3 presents *OntologyToUI* as a Smart_LEM platform module that extracts information from ontologies and generates specifications of the UI

elements composing the interface, detailing the vocabulary of UI element types and the five-ontology architecture. Section 4 presents the rendering architecture. Section 5 validates the framework through two case studies. Section 6 concludes and discusses future work.

2 State of the Art

In this section, several works are contextualized and related with the proposed objectives, covering domain-specific languages and model-driven engineering, model-based user interface development, and the use of ontologies as a modelling substrate.

2.1 DSLs and Model-Driven Engineering

Mernik, Heering and Sloane [7] provide a systematic taxonomy of when and how DSLs should be developed, identifying ontology-based representations as a viable alternative to grammar-based external DSLs. A key advantage of DSLs is ease of use: a domain expert can describe system behaviour using domain vocabulary without needing to understand the underlying implementation. Brambilla, Cabot and Wimmer [2] argue that raising the abstraction level from code to models reduces redundancy and the risk of inconsistency across software layers, making MDE particularly valuable in domains with frequent domain evolution.

2.2 Model-Based User Interface Development

The Unifying Reference Framework of Calvary et al. [3] organises UI specification across four abstraction levels (task model, abstract UI, concrete UI, final UI) and remains the dominant conceptual model for multi-target UI generation. MARIA [9] and UsiXML [3] are concrete realisations: Extensible Markup Language (XML)-based DSLs with their own toolchains supporting multi-level transformation. Ruiz, Serral and Snoeck [10] evaluate model-driven UI generation approaches and note that most systems require proprietary notation, do not scale easily to evolving domains, and address the UI layer in isolation without co-designing the backend from the same model.

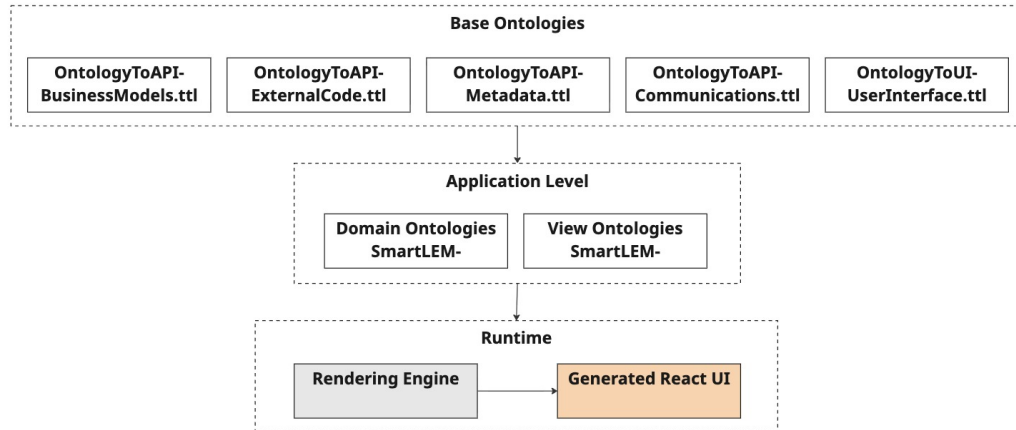
2.3 Ontologies as a Modelling Substrate

Gruber [5] defines an ontology as a formal, explicit specification of a shared conceptualisation. The OWL 2 specification [8] and SPARQL provide machine-processable semantics and a standardised query mechanism over RDF graphs. Prior work has used ontologies to annotate existing UI elements [1], but to our knowledge no published system uses a view-level OWL ontology as a directly executable UI specification interpreted by a generic runtime engine that hardcodes no element types, no domain operations, and no assumptions about which views exist, rendering only what the view ontology declares. OntologyToUI addresses this gap by treating the World Wide Web Consortium (W3C) OWL/RDF stack as the modelling substrate, obtaining formal semantics and standardised querying without custom parsers or compilers.

3 The OntologyToUI DSL

In this section, the OntologyToUI DSL is presented, covering its architecture, the DSL vocabulary of UI element types, and how a view ontology is authored.

3.1 Five-Ontology Architecture



■ **Figure 1** Five-ontology architecture of the Smart_LEM framework. Base ontologies define the metamodel vocabulary; application-level ontologies instantiate it for a specific domain; the rendering engine interprets view ontologies at runtime.

The framework is organised in two levels. At the *base level*, five modular OWL 2 ontologies define the abstract vocabulary (Figure 1):

- **OntologyToAPI-Communication:** models seven communication modalities (Representational State Transfer (REST) API, database, file system, message queue, WebSocket, Remote Procedure Call (RPC), service mesh).
- **OntologyToAPI-ExternalCode:** models references to Python functions and their library dependencies.
- **OntologyToAPI-Metadata:** models data fields, types, and source bindings.
- **OntologyToAPI-BusinessModel:** ties operations to input parameters, required metadata, and external code.
- **OntologyToUI-UserInterface:** defines UI element types and their configurable properties.

At the *application level*, *domain ontologies* instantiate the base vocabulary for a specific system, and *view ontologies* compose UI element instances that reference those domain objects.

This paper focuses on the last ontology, **OntologyToUI-UserInterface**, which is dedicated to interface construction. The remaining ontologies are described in detail on the Smart_LEM project webpage [11].

3.2 UI Element Vocabulary

OntologyToUI-UserInterface defines eight concrete UI element types, shown in Table 1. Every element individual must carry a `ui:order` property that controls its rendering position within the view. The rendering engine queries for individuals of each type at runtime; the set of elements present in a view is therefore determined entirely by the ontology.

■ **Table 1** UI element types defined by `OntologyToUI-UserInterface`.

Element type	Key properties	React component
<code>ui:Label</code>	<code>hasText</code> , <code>hasType</code> , <code>hasLevel</code>	<code>LabelComponent</code>
<code>ui:Form</code>	<code>hasTitle</code> , <code>hasBusinessModel</code>	<code>FormComponent</code>
<code>ui:Table</code>	<code>hasBusinessModel</code> , <code>hasColumnMetadata</code>	<code>TableComponent</code>
<code>ui:LinePlot</code>	<code>hasTitle</code> , <code>hasIndependentMetadata</code> ,	<code>GraphComponent</code>
<code>ui:BarPlot</code>	<code>hasDependentMetadata</code> , <code>hasSeriesAlias</code>	<code>GraphComponent</code>
<code>ui:LiveTable</code>	<code>hasTitle</code> , <code>hasWebSocketUrl</code> , <code>hasTopicFilter</code>	<code>LiveTableComponent</code>
<code>ui:Image</code>	<code>hasMetadata</code> , <code>hasBusinessModel</code>	<code>ImageComponent</code>

3.3 Writing a View Ontology

Following Mernik et al.'s taxonomy [7], `OntologyToUI` is a `Smart_LEM` platform module that uses an *ontology-based external DSL* to define the UI elements to be generated: the base ontologies form the grammar, each view ontology is a program, and the rendering engine is the interpreter. Domain experts author programs in Turtle without programming knowledge.

Listing 1 shows a minimal program: a three-element view that generates a form for creating an energy community (a local group of prosumers that collectively share and trade energy). Each `ui:Label` declares `hasType` (heading style: "title" or "paragraph"), `hasText` (the displayed text), and `ui:order` (render position). The `ui:Form` element references a `BusinessModel` individual defined in the domain ontology; the rendering engine resolves the reference at runtime to discover the operation's parameters, Hypertext Transfer Protocol (HTTP) endpoint, and response schema, without any of that information being present in the view ontology.

■ **Listing 1** View ontology for an energy community creation page (`SmartLEM-CreateEC_View.ttl`).

```
@prefix ui:    <http://www.cedri.com/OntologyToUI-UserInterface#> .
@prefix EC:    <http://www.cedri.com/SmartLEM-EC-Operations#> .
@prefix view:  <http://www.cedri.com/SmartLEM-CreateEC-View#> .

view:PageTitle    rdf:type ui:Label ;
  ui:hasType       "title" ;
  ui:hasText       "Create Energy Community" ;
  ui:hasLevel      1 ;
  ui:order         1 .

view:Description  rdf:type ui:Label ;
  ui:hasType       "paragraph" ;
  ui:hasText       "Fill the form to create a new LEM community." ;
  ui:order         2 .

view:CreateECForm rdf:type ui:Form ;
  ui:hasTitle      "Energy Community Details" ;
  ui:hasBusinessModel EC:CreateEnergyCommunity ;
  ui:order         3 .
```

This separation means that the grammar (base ontologies) can be extended with new element types and the interpreter (rendering engine) can be upgraded independently of any existing program (view ontology). These abstractions eliminate the need to write React component code manually: a domain expert authors a Turtle file and the rendering engine generates the interface automatically.

4 Rendering Architecture

This section describes the UI generation process based on the view ontology, covering the three-layer architecture and the Generic Rendering Principle that ensures domain-independent interpretation.

4.1 Three-Layer Design

The system is organised into three layers (Figure 2).

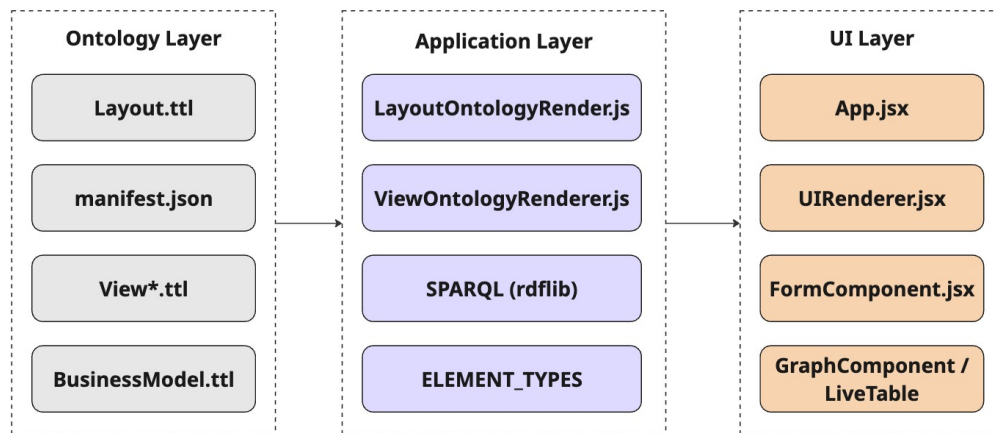


Figure 2 Three-layer architecture of OntologyToUI. Turtle files in the Ontology Layer are parsed into an `rdflib` graph store; SPARQL queries extract element definitions, which are dispatched to React components.

The **Ontology Layer** contains all domain knowledge as Turtle files: a layout ontology defines the page shell; a `manifest.json` lists available views; view ontologies declare UI elements; and domain ontologies define business operations and data sources.

The **Application Layer** is entirely generic. The `rdflib` library (v2.3.5) parses ontology files into an in-memory RDF graph store. `LayoutOntologyRender.js` queries for layout configuration by type without hardcoding any individual Uniform Resource Identifier (URI). `ViewOntologyRender.js` runs one SPARQL SELECT query per element family and assembles a flat list of element descriptors sorted by `ui:order`. Using SPARQL instead of manual triple traversal keeps the extractor declarative and domain-independent: the same queries work over any conformant ontology without custom parsing code.

The **UI Layer** is built on React, Ant Design (v6.0), and Plotly.js. `UIRender.js` maps each element descriptor to a React component through a `COMPONENT_MAP` dispatch table and renders the list in order. No component contains domain knowledge; they depend only on the props passed from the extractor.

4.2 Generic Rendering Principle

The `ELEMENT_TYPES` configuration object in `ViewOntologyRender.js` is the sole coupling point between the grammar (ontology schema) and the interpreter (rendering engine). Each entry specifies the `rdf:type` URI to query for and the ontology properties to extract:

```

LiveTable: {
  type:      UI('LiveTable'),
  properties: {
    title:    UI('hasTitle'),
    wsUrl:    UI('hasWebSocketUrl'),
    topicFilter: UI('hasTopicFilter'),
  },
},

```

Each entry declares the `rdf:type` URI to match and the ontology properties to extract from each individual; the rendering engine uses this configuration to build component descriptors dynamically, with no knowledge of any specific domain. Adding a new element type requires adding one entry to `ELEMENT_TYPES`, one entry to `COMPONENT_MAP`, and creating the corresponding React component; no other file is modified. The current layout model renders elements sequentially, stacked vertically by `ui:order`; this reflects the characteristics of the energy simulation systems targeted by the framework, where forms, tables, and charts are naturally presented as a vertical stack.

5 Case Studies

We validate the framework through two applications from qualitatively different domains and integration paradigms.

5.1 SmartLEM: Local Electricity Market Management

SmartLEM [11] is a business model-oriented platform for developing Local Electricity Markets (LEMs) [6] and accelerating clean energy transition. It provides a digital solution with business models targeting three actor classes: prosumers (participants who both produce and consume energy), aggregators (entities that coordinate groups of prosumers in energy markets), and LEM market participants. The platform supports market-based trading mechanisms, auctions, and settlement procedures, enabling prosumers and communities to trade energy surplus and flexibility (the capacity to adjust energy production or consumption on demand) while increasing the use of renewable energy sources. Its frontend generates all views from ontologies; its backend REST API is likewise generated from the same business model ontologies.

Domain ontologies

SmartLEM-ECModels.ttl defines eleven `bm:BusinessModel` individuals covering community creation, member and asset enrolment, energy simulation, and data retrieval. Each individual is bound to input parameters (types: `str`, `int`, `float`, `UploadFile`), a Python wrapper function, and output metadata individuals that describe the response schema.

View ontologies

Eleven view ontologies were authored, one per application page. Table 2 summarises their distribution across functional categories and element types.

Separation of concerns

All eleven views were created by authoring Turtle files and adding entries to `manifest.json`. The following source files were not modified during the entire development of the SmartLEM views: `App.jsx`, `ViewOntologyRenderer.js`, `UIRenderer.jsx`, `FormComponent.jsx`, `GraphComponent.jsx`, `TableComponent.jsx`. This confirms the Generic Rendering Principle: domain experts can author new views in Turtle without touching application code.

■ **Table 2** SmartLEM view ontologies grouped by functional category.

Category	Views	UI Elements Used
Community Management	CreateEC, GetEC	Label, Form
Member & Asset Mgmt	AddMember, AddAsset, GetMember	Label, Form
Simulation	SimulateEC	Label, Form
Data Visualisation	EC ConsProd, EC Market, Member ConsProd, Member Asset, Weather Graph	LinePlot, Table

Figure 3 shows the *Create Energy Community* view, one of the eleven views generated entirely from Turtle ontologies. The interface supports the full lifecycle of a Local Electricity Market: managing communities, members and assets; running energy simulations; and visualising consumption, production, market balance, and weather data.

■ **Figure 3** Screenshot of the SmartLEM *Create Energy Community* view generated by OntologyToUI.

5.2 Smart_LEM: Real-Time IoT Dashboard

This case study demonstrates that the framework handles event-driven, push-based data sources with no REST API layer. The data producer is a Python agent-based smart home simulator that publishes timestamped sensor readings (energy, water, solar, battery, climate) to an MQTT broker (Mosquitto) via two topic patterns:

```
clients/<house_id>/data    and    clients/<house_id>/status
```

The broker exposes a WebSocket listener on port 9001, which is the connection point for the browser-side component.

Extending the framework

None of the existing element types fits the real-time subscription scenario. A new type, `ui:LiveTable`, was introduced by modifying exactly two files (`ELEMENT_TYPES` and `COMPONENT_MAP`) and adding one React component (`LiveTableComponent.jsx`). No other file in the codebase was changed.

View ontology

The complete dashboard specification is given in Listing 2. It declares four individuals: two `ui:Label` elements and two `ui:LiveTable` elements targeting different MQTT topic filters on the same broker endpoint. The view contains no `ui:BusinessModel` reference and no link to any REST ontology; the specification is self-contained in 29 lines of Turtle.

■ **Listing 2** Complete IoT dashboard view ontology (SmartLEM-Dashboard_View.ttl).

```
@prefix ui:    <http://www.cedri.com/OntologyToUI-UserInterface#> .
@prefix view:  <http://www.cedri.com/SmartLEM-Dashboard-View#> .

view:PageTitle    rdf:type ui:Label ;
    ui:hasText     "IoT Client Monitor" ;
    ui:hasType     "title" ; ui:hasLevel 1 ; ui:order 1 .

view:SubTitle     rdf:type ui:Label ;
    ui:hasText     "Real-time sensor data via MQTT" ;
    ui:hasType     "paragraph" ; ui:order 2 .

view:ClientDataTable    rdf:type ui:LiveTable ;
    ui:hasTitle         "Live Sensor Data" ;
    ui:hasWebSocketUrl  "ws://127.0.0.1:9001" ;
    ui:hasTopicFilter   "clients/+/data" ;
    ui:order            3 .

view:ClientStatusTable  rdf:type ui:LiveTable ;
    ui:hasTitle         "Client Status" ;
    ui:hasWebSocketUrl  "ws://127.0.0.1:9001" ;
    ui:hasTopicFilter   "clients/+/status" ;
    ui:order            4 .
```

LiveTableComponent

The React component connects to the broker using the MQTT protocol over WebSocket via the `mqtt` npm package, subscribes to the configured topic filter, and for each incoming message appends the parsed JavaScript Object Notation (JSON) payload, augmented with a client identifier and a timestamp, to a sliding window of at most 20 rows. *Schema self-discovery*: columns are computed at render time by collecting all keys present across received payloads, so the table columns are inferred automatically from the received data, with no schema declaration required in the view ontology.

Integration

This view is registered in `manifest.json` alongside the eleven SmartLEM views. From the application’s perspective the view is indistinguishable from any other: it is discovered, loaded, parsed, and rendered by the same pipeline.

Figure 4 shows the resulting dashboard. Two live tables update automatically as sensor data arrives: one displays real-time climate and energy readings per client, and the other shows the connection status of each smart home.

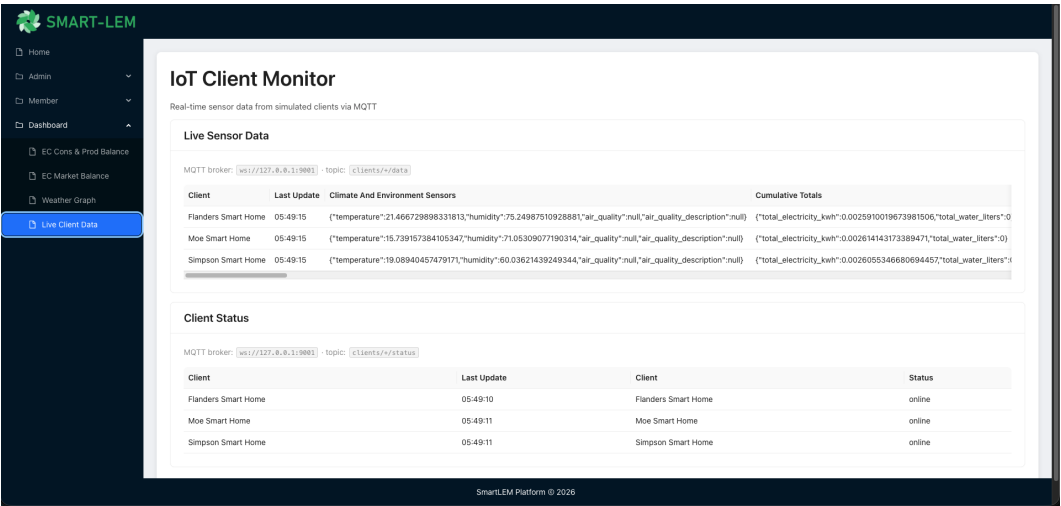


Figure 4 Screenshot of the IoT dashboard generated by OntologyToUI, showing real-time sensor data received via MQTT.

6 Conclusion

We presented OntologyToUI, an ontology-based DSL framework for automatic user interface generation. The Smart_LEM five base ontologies define a reusable metamodel; view ontologies written in Turtle serve as declarative UI programs; and a generic React rendering engine interprets them through SPARQL without domain-specific logic.

Two case studies validate complementary properties of the framework. SmartLEM validates the Generic Rendering Principle: eleven views of four element-type families were created without modifying any application source file, confirming clean separation between domain knowledge, rendering logic, and presentation. The IoT dashboard case study validates extensibility: a new real-time MQTT-subscribed element type was added by changing two files and adding one component, while the rest of the codebase remained intact. Together, the case studies confirm that the framework is domain-independent, spanning REST API-connected UIs and event-driven push-based UIs.

Future work

Shapes Constraint Language (SHACL) validation shapes would allow structural errors in view ontologies to be reported before rendering. A visual ontology editor, where domain experts compose views by dragging and dropping element types onto a canvas, would generate the corresponding Turtle automatically, removing the need for RDF syntax knowledge. A quantitative evaluation comparing lines of Turtle against lines of equivalent handwritten React code would provide stronger empirical evidence of authoring effort reduction.

References

- 1 Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001. doi:10.1038/scientificamerican0501-34.
- 2 Marco Brambilla, Jordi Cabot, and Manuel Wimmer. *Model-Driven Software Engineering in Practice*. Morgan & Claypool, 2 edition, 2017.
- 3 Gaëlle Calvary, Joëlle Coutaz, David Thevenin, Quentin Limbourg, Laurent Bouillon, and Jean Vanderdonckt. A unifying reference framework for multi-target user interfaces. *Interacting with Computers*, 15(3):289–308, 2003. doi:10.1016/S0953-5438(03)00010-9.
- 4 Daniel Theisges da Costa, Júlio Costa, Paulo Jorge Teixeira Matos, and Maria João Tinoco Varanda Pereira. SmartLEM / OntologyToUI. Software, version v1.0., swhId: swh:1:dir:840123e5e0c6e231e56bad1d41fe75ba1890f803 (visited on 2026-07-17). URL: <https://github.com/NewDaniC/OntologyToUI>, doi:10.4230/artifacts.27314.
- 5 Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993. doi:10.1006/knac.1993.1008.
- 6 Esther Mengelkamp, Johannes Gärttner, Kerstin Rock, Scott Kessler, Lawrence Orsini, and Christof Weinhardt. Designing microgrid energy markets: A case study: The Brooklyn Microgrid. *Applied Energy*, 210:870–880, 2018. doi:10.1016/j.apenergy.2017.06.054.
- 7 Marjan Mernik, Jan Heering, and Anthony M. Sloane. When and how to develop domain-specific languages. *ACM Computing Surveys*, 37(4):316–344, 2005. doi:10.1145/1118890.1118892.
- 8 Boris Motik, Peter F. Patel-Schneider, and Bijan Parsia. OWL 2 web ontology language: Structural specification and functional-style syntax. W3c recommendation, W3C, 2012. URL: <https://www.w3.org/TR/owl2-syntax/>.
- 9 Fabio Paternò, Carmen Santoro, and Lucio Davide Spano. MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. *ACM Transactions on Computer-Human Interaction*, 16(4), 2009. doi:10.1145/1614390.1614394.
- 10 Jenny Ruiz, Estefanía Serral, and Monique Snoeck. Evaluating user interface generation approaches: Model-based versus model-driven development. *Software & Systems Modeling*, 18:2753–2776, 2019. doi:10.1007/s10270-018-0698-x.
- 11 Smart_LEM Consortium. Smart local energy market (Smart_LEM) project, 2022. Accessed: April 2026. URL: <https://smartlem.ase.ro/>.

A Silent Multi-Agent Recommendation Engine with Graph-Augmented Memory for Immersive B2B E-Commerce

Davi Silva Eleuterio ✉ 

CeDRI, SusTEC, Instituto Politécnico de Bragança, Portugal

Pedro Filipe Oliveira ✉ 

CeDRI, SusTEC, Instituto Politécnico de Bragança, Portugal

Paulo Matos ✉ 

CeDRI, SusTEC, Instituto Politécnico de Bragança, Portugal

Abstract

Currently, e-commerce platforms are transitioning from static 2D catalogues to immersive environments, requiring recommendation systems to process complex behavioural signals. Traditional collaborative filtering struggles with cold-start scenarios and lacks semantic understanding of product catalogues. This paper presents a *silent* multi-agent recommendation engine developed for the VIMOS (Virtual Integrated Market Online Shopping) project, designed to operate without explicit textual input. Orchestrated via a LangGraph Directed Acyclic Graph, the system utilises Agentic Retrieval-Augmented Generation (RAG) to dynamically synthesise behavioural signals, dense vector retrieval (Qdrant), and knowledge graph traversal (Apache AGE). A locally hosted Llama 3.1 8B model autonomously extracts implicit user preferences to update a persistent graph database and generates deterministic, human-readable explanations for each recommendation. A two-shot verifier engineering pattern is introduced to resolve the structural incompatibility between dynamic candidate sets and static graph execution. The architecture is validated using a controlled synthetic dataset of B2B industrial products – sufficient for functional validation of all agent routing paths – demonstrating high retrieval accuracy, strict constraint adherence (stock and availability verification), and sub-4-second end-to-end latency (when the generative semantic justification step is bypassed) with no external API dependencies.

2012 ACM Subject Classification Applied computing → Document analysis; Computing methodologies → Information extraction; Computing methodologies → Machine learning

Keywords and phrases multi-agent systems, agentic RAG, product recommendation, LangGraph

Digital Object Identifier 10.4230/OASISs.SLATE.2026.15

Funding This work was supported by FCT – Fundação para a Ciência e Tecnologia, I.P. by projects: CeDRI, UID/05757/2025 (DOI: <https://doi.org/10.54499/UID/05757/2025>) and UID/-PRR/05757/2025 (DOI: <https://doi.org/10.54499/UID/PRR/05757/2025>); SusTEC, LA/P/0007/2020 (DOI: <https://doi.org/10.54499/LA/P/0007/2020>) and by NORTE2030-FEDER-02214300 – “VIMOS – Virtual Integrated Market Online Shopping”, through the Fundo Europeu de Desenvolvimento Regional (FEDER) and Programa Regional do Norte (Norte 2030) do Portugal 2030.

 NORTE
Programa Regional do Norte

 PORTUGAL
2030

 Cohesion for the
European Future

1 Introduction

The contemporary e-commerce landscape is undergoing a profound structural transition, shifting from traditional, static 2D product catalogues toward immersive, spatial shopping environments leveraging 3D visualisation and Augmented Reality (AR) [10]. This shift is particularly critical in the Business-to-Business (B2B) sector – such as the dental and industrial tool markets – where products require strict technical matching, precise compatibility checks,



© Davi Silva Eleuterio, Pedro Filipe Oliveira, and Paulo Matos;
licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 15; pp. 15:1–15:14

OpenAccess Series in Informatics



OASIS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and a deep understanding of complex specifications. While conventional recommendation systems rely heavily on standard collaborative filtering, these legacy approaches often struggle with cold-start scenarios and lack the semantic reasoning required to understand the intricate features of specialised B2B catalogues.

Simultaneously, the field of Artificial Intelligence has seen a paradigm shift driven by Large Language Models (LLMs) and the evolution of Retrieval-Augmented Generation (RAG) architectures [6]. By grounding LLMs in external, up-to-date knowledge bases, RAG mitigates hallucinations and enables deep semantic reasoning over technical documents – a paradigm currently evolving into Agentic RAG architectures where autonomous agents orchestrate complex retrieval workflows [14]. However, integrating these conversational agents into an immersive 3D shopping experience presents a user experience challenge: forcing a user to interrupt their spatial exploration to type explicit queries can break immersion and introduce friction. There is a pressing need for *silent* recommendation systems that can operate implicitly, leveraging rich behavioural signals – such as the time spent analysing a 3D mesh or interacting with a product hotspot – without requiring any conversational input.

To address these challenges, this paper presents the design and implementation of a silent multi-agent product recommendation assistant developed within the scope of the Virtual Integrated Market Online Shopping (VIMOS) project. VIMOS aims to replace static B2B e-commerce with a smart 3D visualisation platform driven by Agentic RAG. Three specialised assistants are planned for the platform: the silent recommendation engine described in this paper, a semantic product search assistant, and a 3D spatial context assistant. Operating entirely in the background during page loads, the proposed architecture is orchestrated via LangGraph, utilising a multi-agent system to autonomously evaluate user profile sufficiency [12]. It blends deterministic collaborative filtering with semantic vector search. Utilising a locally hosted Llama 3.1 8B model, the system not only verifies the strict availability of candidate products but also synthesises deterministic, human-readable semantic justifications for each recommendation. Furthermore, it asynchronously updates a persistent Apache AGE knowledge graph to maintain a long-term memory of implicit user preferences.

This work builds upon a line of research by the same group on LLM-powered assistants for specialised domains [5], now extending the scope from reactive conversational RAG to a proactive, graph-augmented, fully local multi-agent pipeline.

The primary contributions of this work are:

1. A fully implemented LangGraph Directed Acyclic Graph for non-conversational, silent product recommendation from purely behavioural signals.
2. The *two-shot verifier pattern*: a practical engineering solution tailored to state initialization constraints in LangGraph pipelines, enabling post-retrieval validation when candidate identifiers are unknown at graph construction time.
3. A session-adaptive four-mode routing strategy guaranteeing non-empty recommendations across authenticated, partially profiled, anonymous, and cold-start sessions.
4. An asynchronous Memory Agent that writes user preference edges to an Apache AGE knowledge graph without adding to response latency.
5. Experimental validation of five functional criteria across 20 synthetic B2B test sessions, achieving 100% product availability compliance and sub-4-second end-to-end latency.

The remainder of this paper is organised as follows. Section 2 reviews related work on RAG evolution and recommender systems. Section 3 details the multi-agent topology, including behavioural preference scoring and session routing. Section 4 discusses the Verifier Agent and the asynchronous Memory Agent. Section 5 presents the experimental setup, synthetic dataset, validation results, and latency analysis. Finally, Section 6 outlines conclusions and future directions.

2 Related Work

The architecture of the proposed recommendation engine lies at the intersection of evolving e-commerce recommender systems, advancements in Large Language Models (LLMs), and the paradigm shift toward Multi-Agent Retrieval-Augmented Generation (Agentic RAG).

2.1 Evolution of Recommendation Systems: From Filtering to Semantic Search

Traditional e-commerce platforms have historically relied on standard Collaborative Filtering (CF) and Content-Based Filtering. These legacy systems were heavily dependent on static demographic segmentation, identifying users with overlapping purchase histories to propagate preferences [7]. However, as digital commerce transitions towards immersive, spatial environments [10], the state of the art has fundamentally shifted toward real-time behavioural analysis.

To handle this increased complexity, modern systems have moved beyond basic Natural Language Processing (NLP) rules [11]. Contemporary hybrid architectures combine traditional paradigms with deep learning techniques and session-level signals (e.g., dwell time, 3D interaction type). The integration of dense vector embeddings into these pipelines represents the most recent evolution: a pre-trained embedding model encodes both product descriptions and behavioural patterns into a shared semantic space, enabling retrieval by semantic proximity. The present work adopts this advanced hybrid paradigm, merging graph-based collaborative filtering via Apache AGE with dense vector search over Qdrant.

2.2 Agentic RAG and LLMs in Information Retrieval

To overcome the semantic limitations of traditional recommender systems, Retrieval-Augmented Generation (RAG) became the industry standard to ground LLMs in external knowledge. RAG architectures have evolved through four recognised generations: Naive RAG, Advanced RAG, Modular RAG, and Agentic RAG [6, 15, 14]. Early iterations (“Naive RAG”) operated on strictly linear, retrieve-then-read pipelines, while “Advanced RAG” injected pre- and post-retrieval optimisations such as query rewriting and cross-encoder reranking.

Recently, there has been a profound paradigm shift towards “Agentic RAG”. In this architecture, LLMs transition from fast, linear “System 1” processing to autonomous “System 2” cognitive reasoning [8]. Elevating the LLM to an autonomous orchestrator operating within a ReAct (Reasoning and Acting) loop, these systems dynamically route across heterogeneous data sources. While recent implementations of vanilla RAG have demonstrated high retrieval success in reactive, conversational interfaces for specialised domains [5], the field is rapidly advancing. The current frontier demands a shift from these reactive conversational pipelines to proactive, graph-augmented Multi-Agent Systems (MAS) capable of interpreting implicit, non-verbal behavioural signals.

2.3 Multi-Agent Orchestration and Deterministic Logic

The distinction between a single agentic loop and a true MAS is formalised by Sapkota et al. [13], who define MAS as a system where multiple specialised sub-agents collaborate, decompose tasks, and cross-verify outputs. This topology is crucial for enforcing deterministic business logic and actively identifying hallucinations in technical domains [12].

A prominent architectural pattern that has emerged to improve efficiency is **Adaptive RAG**, which interposes a complexity classification logic that routes queries dynamically, bypassing heavy agentic pipelines for simple factual queries to optimise global latency. The

present work adopts a related principle: the Orchestrator Agent deterministically routes sessions by profile completeness, avoiding unnecessary computation for anonymous or cold-start sessions.

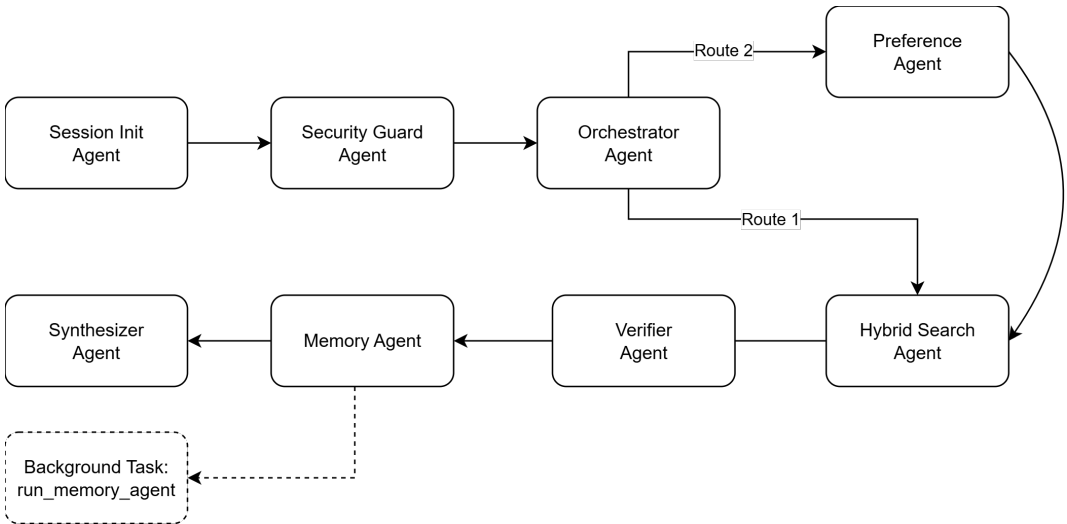
To safely orchestrate these stateful pipelines, modern architectures rely on state graph frameworks such as LangGraph [1]. LangGraph is a Python library built on top of LangChain that models an agent pipeline as a *directed acyclic graph* (DAG): nodes represent individual processing steps (agents or tools), directed edges encode execution order, and conditional edges implement branching logic at runtime. A shared typed state dictionary (**AgentState**) flows through every node, allowing agents to read from and write to a common data structure without direct coupling. This explicit graph abstraction allows for deterministic routing rules alongside probabilistic LLM reasoning, guaranteeing robust and explainable recommendations.

3 System Architecture

This section describes the multi-agent pipeline that powers the silent recommendation engine. The pipeline is composed of eight specialised agents, each responsible for a distinct processing step, coordinated by a shared execution graph. Section 3.4 details how raw behavioural signals are transformed into preference vectors; Section 3.5 describes how those vectors drive hybrid retrieval across four session modes. The infrastructure components (vector store, relational database, knowledge graph, and local LLM runtime) that underpin all agents are introduced in Section 3.1.

3.1 Multi-Agent Topology and Shared State

The recommendation pipeline is implemented as an eight-agent directed acyclic LangGraph StateGraph backed by a shared **AgentState** TypedDict that flows through every agent. Figure 1 illustrates the complete topology and the two routing paths.



■ **Figure 1** Silent recommendation pipeline as a directed acyclic graph of eight agents. Each node represents one agent; directed edges encode execution order; the conditional fork at the Orchestrator Agent implements the two-route logic. The five infrastructure components (FastAPI, Qdrant, PostgreSQL, Apache AGE, Ollama) are shared across all agents and summarised in Table 1. Route 1 (cold-start or insufficient profile) bypasses the Preference Agent and enters directly at the Hybrid Search Agent. The Memory Agent dispatches a non-blocking background task after the HTTP response is delivered (dashed arrow), adding zero latency to the critical path.

The infrastructure layer supporting all agents comprises five components: (i) a FastAPI application (a high-performance Python web framework) with an asyncpg connection pool for non-blocking database access; (ii) Qdrant as the vector database – a dedicated engine for approximate nearest-neighbour search – hosting the `product_vectors` collection of 1 024-dimensional `bge-m3` (BGE Multilingual Embedding model) embeddings indexed with HNSW (Hierarchical Navigable Small World graphs, providing logarithmic-time approximate search); (iii) PostgreSQL as the relational database for purchase history, interaction signals, and product metadata; (iv) Apache AGE (A Graph Extension) as the property graph layer deployed on the same PostgreSQL instance, enabling Cypher-language graph queries for collaborative filtering edges and user preference graphs; and (v) Ollama as the local model-serving runtime for both the `bge-m3` embedding model and the Llama 3.1 8B generative model. Table 1 summarises the stack. Running all inference locally eliminates external API dependencies and satisfies B2B data privacy requirements [4]. The 4-bit quantisation of Llama 3.1 8B reduces VRAM requirements to approximately 5 GB while retaining over 96% of full-precision instruction-following quality [3].

■ **Table 1** Technology Stack.

Component	Technology
Orchestration	LangGraph StateGraph (DAG, 8 agents)
API layer	FastAPI + asyncpg connection pool
LLM inference	Ollama – <code>llama3.1:8b-instruct-q4_K_M</code>
Embeddings	<code>bge-m3</code> (1 024 dims) via Ollama
Vector database	Qdrant (<code>product_vectors</code> collection)
Relational database	PostgreSQL (asyncpg)
Knowledge graph	Apache AGE (property graph on PostgreSQL)
Observability	LangSmith (optional distributed tracing)
Demo interface	Streamlit

3.2 Session Init and Security Guard Agents

The Session Init Agent reads the session hash, authentication state, and language code from the incoming request payload, initialising the shared `AgentState`. The Security Guard Agent applies pattern-matching screening for prompt injection tokens and logs every request to the observability layer. Full adversarial detection logic is deferred to the post-prototype phase; the stub ensures that the guard position is architecturally established from the outset and can be progressively hardened without modifying downstream agents. Furthermore, it is important to note that the current topology deliberately omits a Corrective RAG (CRAG) Evaluator node. This component acts as a placeholder in the overarching VIMOS architecture, intended for future integration to actively grade retrieval relevance and trigger fallback searches if vector candidates fall below a semantic similarity threshold. Documenting its absence clarifies the intended scope of the current pipeline.

3.3 Orchestrator Agent and Session Routing

The Orchestrator Agent applies a two-condition routing decision. If the session is anonymous *or* the authenticated user’s behavioural signal count falls below a minimum completeness threshold, the graph follows Route 1 – entering directly at the Hybrid Search Agent with a pre-built generic query. Otherwise, Route 2 is taken, passing through the Preference Agent and the Hybrid Search Agent. This routing decision is fully deterministic and guarantees that every session exits the pipeline with a non-empty recommendation list – a design invariant enforced by the four session modes described in Section 3.5.

3.4 Preference Agent and Behavioural Scoring

For authenticated sessions with sufficient behavioural history, the Preference Agent computes weighted (category, brand) affinity scores from two relational sources: purchase history records and interaction signals.

Purchase history contributes a fixed weight of 5.0 per completed transaction. Interaction signals span six event types – view, 3D model open, AR view, wishlist add, cart add, and purchase – each assigned a base weight w_t and scaled by a temporal decay factor τ :

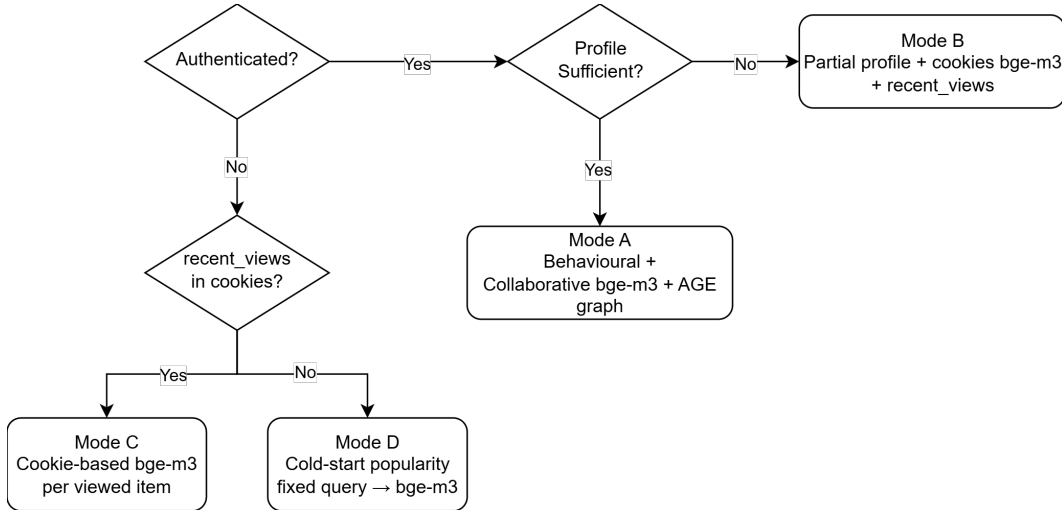
$$\text{score}(e) = w_t \times \tau, \quad \tau = \min\left(\frac{t_{\text{spent}}}{300}, 1.0\right)$$

where t_{spent} is the session dwell time in seconds (capped at 300). The weight hierarchy – view ($w=0.5$) through purchase ($w=5.0$) – reflects purchase intent: 3D and AR interactions carry elevated signal value relative to passive views, consistent with findings on augmented reality engagement in e-commerce [10, 2]. Scores are aggregated per (category, brand) pair across all historical events, and the top-3 pairs by cumulative score are emitted as the `preference_profile` for the Hybrid Search Agent.

This deterministic scoring approach treats the stream of $\{\text{event\_type}, \text{time\_spent}, \text{timestamp}\}$ tuples as an *implicit query language* – a non-textual signal vocabulary subsequently projected into the same semantic embedding space used for natural language product queries. The novelty lies in applying LLM embedding machinery to behavioural telemetry rather than text, positioning the whole system as a language-processing pipeline operating on a structured behavioural signal language rather than a conventional recommendation algorithm.

3.5 Hybrid Search Agent and Session Modes

The Hybrid Search Agent implements four session modes covering the full authentication and profile-completeness space. Figure 2 illustrates the decision logic.



■ **Figure 2** Four session routing modes of the Hybrid Search Agent. Every mode guarantees a non-empty candidate set. All candidates carry $\{\text{product\_id}, \text{score}, \text{source}\}$, where $\text{source} \in \{\text{behavioural}, \text{collaborative}, \text{cookies}, \text{popular}\}$.

Mode A – Authenticated, full profile. For each of the top-3 (category, brand) pairs from the Preference Agent, a synthetic textual query is constructed and embedded with bge-m3, then submitted to Qdrant (top-10 per pair). Concurrently, a Cypher query is executed over Apache AGE to retrieve collaborative filtering candidates: users sharing at least three purchases with the current user whose additionally purchased products are not yet in the current user’s history. Semantic and collaborative candidates are merged and tagged accordingly.

Mode B – Authenticated, insufficient profile. Dense embedding search proceeds with available partial preference data; `recent_views` cookie values complement the candidate set.

Mode C – Anonymous with `recent_views`. Each product identifier in the cookie is embedded individually with bge-m3; the top-10 results per item are merged and tagged cookies.

Mode D – Anonymous cold start. A fixed popularity query is embedded and submitted to Qdrant; all candidates are tagged `popular`.

4 Verification and Memory

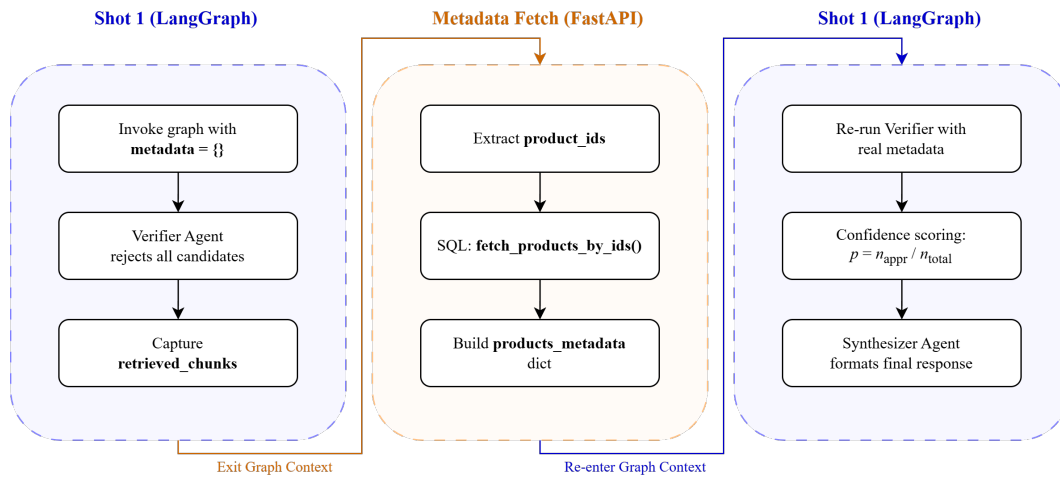
This section details the two components that run at the boundary between retrieval and response delivery. The Verifier Agent (Section 4.1) enforces strict business constraints – stock availability and confidence thresholds – and introduces a novel engineering pattern to overcome a structural limitation of stateful graph pipelines. The Memory Agent (Section 4.2) asynchronously persists inferred user preferences to the knowledge graph after the HTTP response has already been delivered, ensuring zero impact on response latency.

4.1 The Two-Shot Verifier Pattern

The Verifier Agent implements a practical engineering solution to a specific state constraint inherent to LangGraph-based agentic RAG pipelines: *the products metadata dictionary cannot be pre-loaded into the graph at construction time because the candidate product identifiers are only known after the retrieval agents have executed*. Loading metadata for all possible products at graph initialisation is impractical at catalogue scale. While executing a targeted SQL fetch within a designated LangGraph node is theoretically possible, the graph’s static `AgentState` schema typically requires pre-defined structures that complicate dynamic metadata injection for an arbitrary and unpredictable set of candidate identifiers at runtime. The two-shot pattern bypasses this limitation.

The two-shot pattern resolves this by executing the pipeline twice around the verifier boundary, entirely outside the LangGraph execution engine, in the FastAPI endpoint handler. Figure 3 illustrates the full flow.

The Verifier Agent applies a three-tier confidence scoring model. Let $p = n_{\text{approved}}/n_{\text{total}}$ be the fraction of candidates passing stock and availability validation (`stock > 0 AND state = 5`). If $p \geq \alpha = 0.75$, the response is emitted directly with no annotation. If $\beta \leq p < \alpha$ ($\beta = 0.50$), a `verifier_warning` flag is attached to the response. If $p < \beta$, an `escalation_flag` is set for manual review. In production mode ($\text{GPU} \geq 12 \text{ GB VRAM}$), the Verifier Agent additionally invokes Llama 3.1 8B to generate a 15-word semantic justification per approved product at temperature 0.0, producing deterministic, human-readable explanations. In the development environment described here this LLM step is disabled due to shared VRAM constraints between bge-m3 and the language model.



■ **Figure 3** The two-shot verifier pattern. Shot 1 uses the LangGraph graph purely for retrieval; the Verifier Agent’s intentional rejection exposes the candidate `product_ids`. After a targeted SQL metadata fetch, Shot 2 re-executes only the Verifier and Synthesizer Agents outside the graph, with real product metadata. This pattern is reusable in any LangGraph pipeline requiring post-retrieval metadata validation.

4.2 Memory Agent and Asynchronous Graph Writing

The Memory Agent node within the LangGraph graph is a pass-through. Its actual logic executes *after* the HTTP response has been delivered to the client via `asyncio.create_task()`, ensuring zero added latency. The background task executes three sequential steps. First, it records the recommendation event in the relational database. Second, it invokes Llama 3.1 8B with an entity extraction prompt to identify {categories, brands} from the recommendation context. Third, it writes to the Apache AGE property graph, creating or reinforcing directed `(user)-[:INTERESTED_IN]->(category)` and `(user)-[:INTERESTED_IN]->(brand)` edges weighted by the recommendation source type. A `CONTEXT_SHIFT` heuristic flags sudden divergences between new preference signals and the user’s established graph profile, recording them as potential persona changes rather than overwriting historical preferences.

Failure of the background task is logged but does not affect the delivered recommendation: the design invariant is that graph memory enriches future sessions without threatening the current one. This non-blocking pattern ensures that even under hardware constraints, the knowledge graph progressively accumulates richer preference structure across user sessions.

4.3 Synthesizer Agent and Response Format

The Synthesizer Agent is the final node in the LangGraph pipeline and is responsible for serialising the validated recommendation set into a structured API response. Its output carries: (i) the anonymised session identifier; (ii) an ordered list of product recommendations, where each entry includes the product identifier, name, category, similarity score, recommendation source tag (`behavioural`, `collaborative`, `cookies`, or `popular`), semantic justification (when the LLM step is active), and current stock status; and (iii) the end-to-end pipeline latency in milliseconds. This structured format is consumed directly by the VIMOS frontend to render source-coded recommendation cards, as illustrated in Figures 4–6.

5 Experimental Setup and Validation

This section describes the experimental context used to validate the proposed architecture. The evaluation targets functional correctness rather than large-scale benchmarking: given the prototype scope of the VIMOS project, the goal is to confirm that all pipeline components behave as specified across every session mode, and that the end-to-end latency is compatible with real-world deployment.

The case study is a B2B dental and industrial tools marketplace prototype. A Streamlit-based demonstration interface exposes all four session modes of the pipeline (authenticated with full profile, authenticated with partial profile, anonymous with cookie history, and anonymous cold-start) and connects directly to the FastAPI backend. All components – the vector store, relational database, knowledge graph, and LLM runtime – run locally on a single development machine, with no external API calls. Screenshots from this interface illustrate representative outputs for each session mode (Figures 4–6).

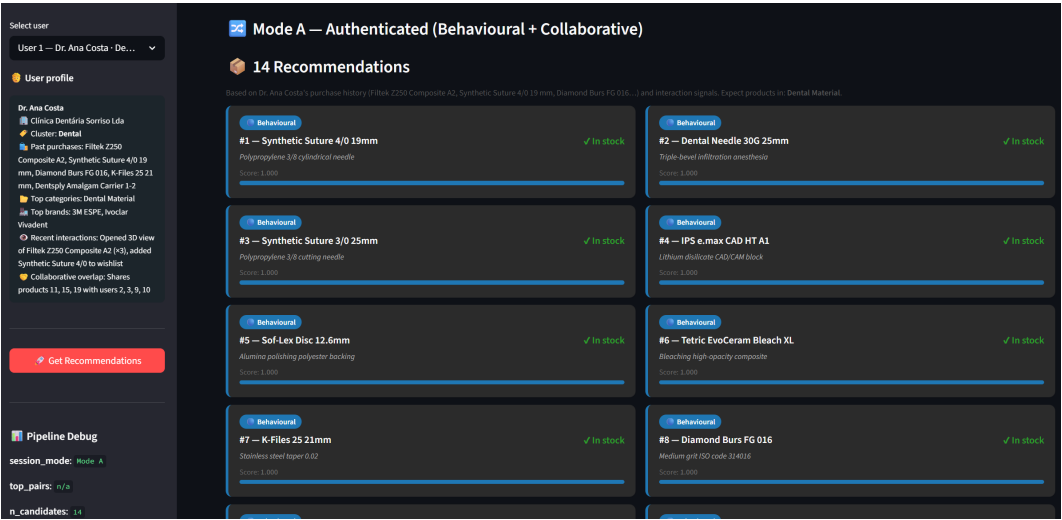


Figure 4 Demo interface: authenticated user with full behavioural profile (Mode A). Recommendation cards are annotated with colour-coded source badges – blue indicates a *behavioural* candidate (derived from the user’s own interaction history) and green indicates a *collaborative* candidate (derived from similar users’ purchase histories) – alongside a numerical similarity score and an in-stock indicator.

5.1 Synthetic B2B Dataset

A synthetic dataset was seeded into PostgreSQL and indexed into Qdrant to enable reproducible validation independent of live production data. The dataset comprises 40 products across four B2B categories: construction tools (brands Bosch, DeWalt, Makita), dental supplies (brands 3M, KaVo, Ivoclar), hospital and medical equipment, and lifestyle goods. Product descriptions were generated as PDF documents and ingested via a synchronisation pipeline using bge-m3, producing 83 vector points in the `product_vectors` Qdrant collection.

The user base comprises 25 synthetic users grouped into three purchase clusters: a dental cluster of 9 users, a construction cluster of 6 users, and a hospital cluster of 5 users. Over 100 interaction records spanning all six interaction types were seeded. The Apache AGE graph was populated with structural and semantic edges providing collaborative filtering paths between users with shared purchase history.

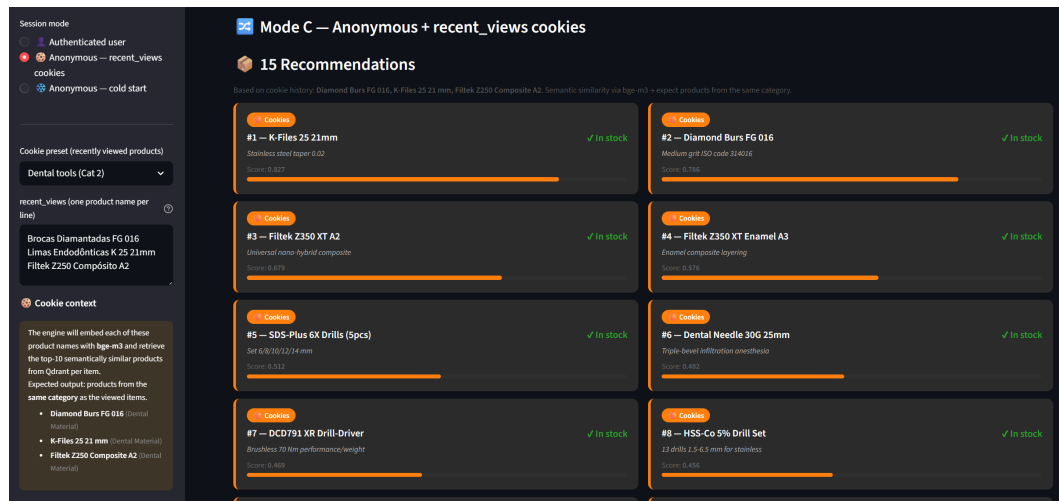
15:10 A Silent Multi-Agent Recommendation Engine for Immersive B2B E-Commerce

While the current scale of 40 products and 25 users represents a limitation for evaluating large-scale retrieval performance, it is sufficient for the functional validation of the agent routing logic presented here. Future scaling to thousands of technical B2B products is structurally supported by the underlying infrastructure: Qdrant’s HNSW index guarantees logarithmic search time complexity $\mathcal{O}(\log N)$, and Apache AGE’s indexed property traversals ensure that collaborative filtering performance remains stable as the user graph expands.

5.2 Demonstration Interface

A Streamlit application (a Python-based rapid web UI framework) was developed to demonstrate all session modes and to generate the screenshot figures presented below. It calls the FastAPI endpoint directly and displays colour-coded source badges (blue for behavioural, green for collaborative, orange for cookies, grey for popular), similarity score bars, stock status indicators, and a pipeline debug sidebar showing the active session mode, computed preference pairs, and candidate count.

Figure 4 shows an authenticated user session with a full behavioural profile, where both behavioural and collaborative source badges are visible. Figure 5 illustrates an anonymous session relying on `recent_views` cookie data. Figure 6 shows a cold-start anonymous session where the popularity fallback fires and all recommendations carry grey source badges.



■ **Figure 5** Demo interface: anonymous user with `recent_views` cookie history (Mode C). Each previously viewed product identifier from the cookie is embedded and used to retrieve semantically similar products; all recommendation cards carry orange *cookies* source badges, reflecting that no authenticated behavioural history is available.

5.3 Validation Protocol and Results

The validation suite executes 20 test sessions covering all four session modes and both authentication states. Five functional criteria were defined:

- C1 – Liveness:** All 20 sessions return HTTP 200 with a non-empty recommendation list.
- C2 – Category Precision:** At least 80% of recommended products for authenticated users belong to the user’s preferred categories as identified by the Preference Agent.

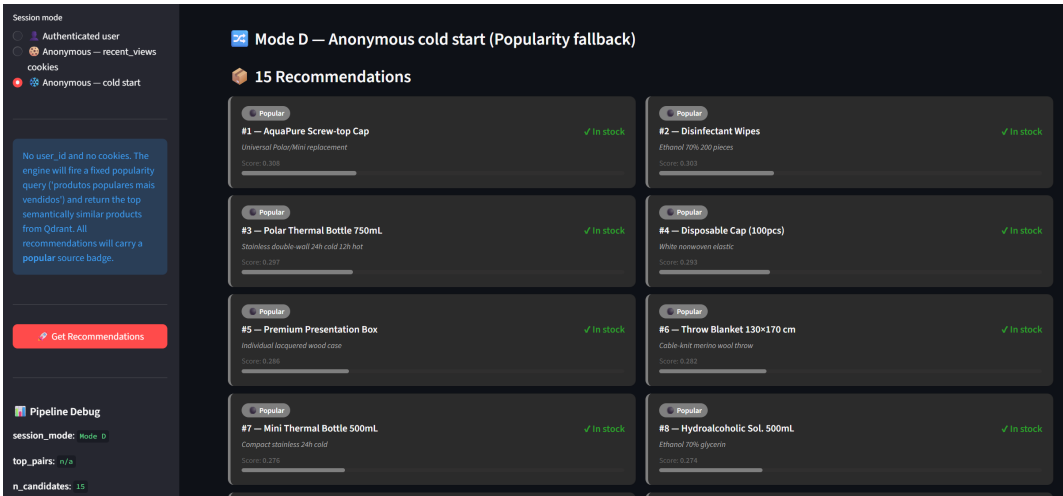


Figure 6 Demo interface: anonymous cold-start session (Mode D). No authenticated profile or cookie history is present; the pipeline falls back to a fixed popularity query submitted to Qdrant. All recommendation cards carry grey *popular* source badges and the session debug panel confirms the *cold_start* mode label.

- C3 – Collaborative Coverage:** At least one collaborative-source candidate appears in the recommendation list for users with sufficient purchase overlap.
- C4 – Stock Compliance:** 100% of recommended products satisfy availability and stock constraints.
- C5 – Latency:** End-to-end response time meets the production target of <4s with warmed models.

Table 2 summarises the outcomes. All five criteria were satisfied. An integration test suite of 10 automated tests also passed in full, covering individual agent behaviour, routing logic, and end-to-end API responses across all session modes.

Table 2 Validation Results – 20 test sessions.

Crit.	Description	Result	Status
C1	HTTP 200 + non-empty list	20/20	Pass
C2	$\geq 80\%$ category precision (auth.)	100%	Pass
C3	≥ 1 collaborative candidate	Confirmed	Pass
C4	100% stock and availability	100%	Pass
C5	Latency <4 s (warm models)	3.4 s avg	Pass

To demonstrate the added value of the multi-agent approach without extensive new benchmarking, a functional comparison against a standard Collaborative Filtering (CF) baseline was analyzed over the synthetic dataset. While a standard CF approach successfully identified candidates for the authenticated user cluster with rich purchase histories (achieving parity with Mode A), it fundamentally failed (0% candidate generation) for anonymous cold-start sessions (Mode D) and recent-views scenarios (Mode C). By contrast, the proposed hybrid multi-agent system guarantees 100% candidate coverage across all session modes by seamlessly orchestrating fallbacks to semantic embedding similarity and popularity metrics.

5.4 Latency Analysis

System latency exhibits significant variance depending on the underlying hardware environment, necessitating a clear distinction between architectural overhead and hardware constraints. In the baseline development configuration (constrained to a GPU with 8 GB of shared VRAM), the bge-m3 embedding model and the Llama 3.1 8B language model are forced into sequential allocation and offloading per request. This VRAM bottleneck produces end-to-end latencies of 6.0 to 8.0 seconds.

However, this is strictly an infrastructural limitation. When deployed on hardware with ≥ 12 GB of VRAM – allowing both models to reside concurrently in memory – the end-to-end execution for an initial session request drops to approximately 3.4 seconds, comfortably satisfying the sub-4-second production target. Profiling the pipeline execution reveals that the dense embedding generation (bge-m3 projecting the synthetic queries) and the subsequent Qdrant vector retrieval dominate the computational overhead. Conversely, the deterministic data retrieval operations – encompassing both standard PostgreSQL relational queries and the Apache AGE Cypher graph traversals – demonstrate high efficiency, contributing a negligible combined latency of less than 100 ms.

Finally, it is important to note that activating the LLM-driven semantic justification within the Verifier Agent (which was disabled during this specific benchmark to preserve memory) introduces an estimated generation overhead of 1.5 to 2.5 seconds per request on dedicated hardware [3]. The decision to enable this feature represents a strict trade-off between explanatory transparency and operational latency. Enabling the LLM step pushes total pipeline execution to approximately 6.0 seconds. While 6 seconds introduces significant friction for standard reactive web interfaces, this latency is highly acceptable – and deliberately accounted for – in the context of the VIMOS project. Because the recommendation engine operates silently in the background while complex 3D scenes and AR assets are loading, the user’s attention is occupied by spatial exploration, effectively masking the generation overhead.

5.5 Cold-Start and Edge Cases

The cold-start anonymous path was exercised in 5 of the 20 sessions; the popularity fallback returned non-empty results in all cases. The anonymous-with-cookies path was tested with simulated `recent_views` payloads, and semantic similarity correctly retrieved products from the same category as the viewed items in every case. The escalation flag ($p < \beta = 0.50$) was not triggered in any session, consistent with the synthetic dataset having complete stock information for all products.

6 Conclusion and Future Work

This paper presented a silent multi-agent product recommendation engine for the VIMOS platform, demonstrating that the proposed agentic RAG pipeline – combining graph-based orchestration, dense semantic vector search, and knowledge-graph collaborative filtering – can effectively power non-conversational, proactive recommendations derived purely from behavioural signals, with all inference running locally and no external API dependencies. The system frames purchase events, interaction type codes, and dwell-time values as an implicit signal language processed through the same LLM embedding infrastructure used for natural language queries.

The primary architectural contribution is the two-shot verifier pattern, a reusable solution to the tension between dynamic candidate generation and static graph execution boundaries that arises in any LangGraph pipeline requiring post-retrieval metadata validation. Session-adaptive four-mode routing ensures non-empty recommendations for every user profile, from fully authenticated buyers to cold-start anonymous visitors. Validation across 20 synthetic B2B test sessions confirms 100% product availability compliance, 100% category precision for authenticated users, and 3.4s end-to-end latency with warmed models. All inference runs locally with no external API dependencies.

Several directions remain for future work. The semantic product search assistant – supporting free-text queries, action-verb contextualisation, and multimodal image upload – and the 3D spatial context assistant – combining Apache AGE multi-hop reasoning with dense retrieval over technical PDF manuals – will complete the VIMOS multi-agent ecosystem. RAGAS evaluation [9] will provide rigorous quantitative assessment of retrieval precision, context recall, and faithfulness for the conversational assistants. Production deployment on a GPU with ≥ 12 GB VRAM will validate the < 4 s latency target with the LLM reasoning step enabled. Apache Airflow will be introduced for scheduled ETL synchronisation between the production catalogue and the Qdrant index. Finally, the two-shot verifier pattern will be formalised as a reusable LangGraph component applicable to any pipeline where retrieval-time candidate identities are required for post-retrieval validation.

References

- 1 Gustavo Aquino, Nádila Azevedo, Leandro Okimoto, Leonardo Camelo, Hendrio Bragça, Rubens Fernandes, André Printes, Fábio Cardoso, Raimundo Gomes, and I. Gondres. From RAG to Multi-Agent Systems: A Survey of Modern Approaches in LLM Development, February 2025. doi:10.20944/preprints202502.0406.v1.
- 2 Mohannad Moufeed Ayyash and Hala Montaser Mohammad Ali. Human–computer interaction meets consumer psychology: How augmented reality shapes decision styles in online fashion purchases. *Computers in Human Behavior Reports*, 21:100985, 2026. doi:10.1016/j.chbr.2026.100985.
- 3 Gautam B and Anupam Purwar. Evaluating the Efficacy of Open-Source LLMs in Enterprise-Specific RAG Systems: A Comparative Study of Performance and Scalability, June 2024. arXiv:2406.11424 [cs]. doi:10.48550/arXiv.2406.11424.
- 4 Wan Chong Choi and Chi In Chang. Advantages and Limitations of Open-Source versus Commercial Large Language Models (LLMs): A Comparative Study of DeepSeek and OpenAI’s ChatGPT, March 2025. doi:10.20944/preprints202503.1081.v1.
- 5 Davi Silva Eleuterio, Pedro Filipe Oliveira, Paulo Matos, and Jorge Manuel Afonso Alves. A chatbot to help promoting financial literacy. In *Proceedings of the 14th Symposium on Languages, Applications and Technologies (SLATE 2025)*, OASICS, 2025. Same authors, prior conference paper. doi:10.4230/OASICS.SLATE.2025.7.
- 6 Mohamed Fezari and Ali Al Dahoud. The Evolution of Retrieval-Augmented Generation (RAG) in AI, January 2026. doi:10.13140/RG.2.2.27107.62245.
- 7 Xiaopeng Li, Pengyue Jia, Derong Xu, Yi Wen, Yingyi Zhang, Wenlin Zhang, Wanyu Wang, Yichao Wang, Zhaocheng Du, Xiangyang Li, Yong Liu, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. A Survey of Personalization: From RAG to Agent, April 2025. arXiv:2504.10147 [cs]. doi:10.48550/arXiv.2504.10147.
- 8 Jintao Liang, Gang Su, Huifeng Lin, You Wu, Rui Zhao, and Ziyue Li. Reasoning RAG via System 1 or System 2: A Survey on Reasoning Agentic Retrieval-Augmented Generation for Industry Challenges, June 2025. arXiv:2506.10408 [cs]. doi:10.48550/arXiv.2506.10408.

- 9 Haitao Lu, Dayu Zheng, and Luo Yang. Mutual information-based retrieval-augmented generation for domain question answering. *Knowledge and Information Systems*, 68:49, January 2026. doi:10.1007/s10115-025-02624-x.
- 10 Merylin Monaro, Alice Bettelli, Giovanni Portello, Leonardo Pierobon, Valeria Orso, Ariel Caputo, Maria Luisa Campanini, Andrea Giachetti, and Luciano Gamberini. Enhancing shopping experience in augmented reality by customizing product manipulation modalities: A customer experience study. *International Journal of Human-Computer Studies*, 202:103553, 2025. doi:10.1016/j.ijhcs.2025.103553.
- 11 Rahul Mundlamuri, Ganesh Reddy Gunnam, Nikhil Kumar Mysari, and Jayakanth Pujuri. The Evolution of AI: From Classical Machine Learning to Modern Large Language Models. *IEEE Access*, 13:178302–178341, 2025. doi:10.1109/ACCESS.2025.3621344.
- 12 R. Pugliese, G. Kourousias, F. Venier, and G. Garlatti Costa. Agentic publications: redesigning scientific publishing in the age of thinking large language models. *Journal of Documentation*, 82(7):125–149, 2026. doi:10.1108/JD-07-2025-0207.
- 13 Ranjan Sapkota, Konstantinos I. Roumeliotis, and Manoj Karkee. AI Agents vs. Agentic AI: A Conceptual taxonomy, applications and challenges. *Information Fusion*, 126:103599, February 2026. doi:10.1016/j.inffus.2025.103599.
- 14 Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG, February 2025. arXiv:2501.09136 [cs]. doi:10.48550/arXiv.2501.09136.
- 15 Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. Retrieval-Augmented Generation for AI-Generated Content: A Survey. *Data Science and Engineering*, January 2026. doi:10.1007/s41019-025-00335-5.

Coordinating Multi-Assistant Interaction in Smart Homes: A Front-End Delegation Approach

Pedro Carneiro ✉ 

IEETA, DETI, University of Aveiro, Portugal

LASI – Intelligent System Associate Laboratory, Guimarães, Portugal

Inês Águia ✉ 

IEETA, DETI, University of Aveiro, Portugal

LASI – Intelligent System Associate Laboratory, Guimarães, Portugal

Nuno Almeida ✉ 

IEETA, DETI, University of Aveiro, Portugal

LASI – Intelligent System Associate Laboratory, Guimarães, Portugal

António Teixeira ✉ 

IEETA, DETI, University of Aveiro, Portugal

LASI – Intelligent System Associate Laboratory, Guimarães, Portugal

Abstract

Smart home environments increasingly integrate multiple domain-specific assistants, yet interaction remains fragmented across heterogeneous interfaces. Users are required to adapt to different systems and interaction modalities, which introduces complexity and reduces accessibility, particularly in scenarios such as active ageing.

This work aims to design a unified conversational abstraction that enables seamless interaction with multiple specialised assistants through a single entry point. The goal is to decouple user interaction from underlying services while preserving modularity and extensibility.

We present the **Butler Assistant**, a modular conversational framework that orchestrates multiple domain-specific assistants within a unified architecture. The system introduces a routing layer based on a shared registry and supports interchangeable routing engines. Two approaches are explored: a lexical model based on Naive Bayes and a semantic model based on sentence embeddings. The system is evaluated under both in-domain and out-of-domain scenarios.

The evaluation shows that the semantic model improves routing accuracy in domains with limited training data, demonstrating stronger generalization capabilities. In contrast, the lexical model is more sensitive to training data distribution, achieving strong performance in well-represented domains but degrading in low-resource scenarios. Importantly, both approaches exhibit reliable behaviour in out-of-domain conditions, consistently redirecting unsupported inputs to fallback outcomes and avoiding incorrect domain assignments. These results highlight the complementary strengths of lexical and semantic routing and emphasize the role of system-level decision mechanisms in ensuring safe and robust conversational interaction.

2012 ACM Subject Classification Human-centered computing → Natural language interfaces; Computing methodologies → Natural language processing; Human-centered computing → Interaction paradigms

Keywords and phrases Conversational Assistants, Smart Homes, Naïve Bayes, Embedding-driven Semantic Similarity

Digital Object Identifier 10.4230/OASICS.SLATE.2026.16

Funding This research was supported by projects Casa Viva+ and VITALITY (COMPETE2030-FEDER-00755100 16385) and Research Unit IEETA, funded by National Funds through the FCT – Foundation for Science and Technology, in the context of the project UIDB/00127/2020.



© Pedro Carneiro, Inês Águia, Nuno Almeida, and António Teixeira;
licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 16; pp. 16:1–16:16

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Conversational assistants now serve as primary interfaces for interacting with smart home environments, providing natural, hands-free access to a wide range of domestic functions. These systems support a range of everyday activities, including control of domotic systems, kitchen assistance, access to contextual information such as weather forecasts, and guidance during physical exercise. This breadth of functionality positions conversational assistants as key enablers of intuitive and pervasive interaction [17, 3]. This relevance is amplified in households with older adults or individuals receiving home health services, where these interfaces can reduce cognitive load, promote autonomy, and facilitate aging in place [18, 13].

Smart homes oriented toward health monitoring increasingly integrate sensing infrastructures, activity recognition, and AI-based feedback mechanisms to support well-being and chronic disease management [19, 9]. When integrated with conversational interaction, these systems can provide accessible and personalized support in healthcare, especially for self-management, monitoring, and patient education [4, 10].

However, current commercial assistants remain confined to proprietary ecosystems such as Alexa, Google Assistant, and Siri. Each is optimized for a narrow set of capabilities and lacks interoperability across heterogeneous devices and services [7, 8].

This parallelism of smart homes, where multiple assistants coexist (e.g., competing wake words leading to misrecognition), exacerbates fragmentation in complex setups that require coordinated control [8]. A challenge manifests acutely in real-world living lab environments. In these innovative scenarios, which simulate daily life with health monitoring, users encounter difficulties with unified dialogue management. This limitation inhibits the effectiveness of aging-in-place support.

To address these limitations, this paper proposes a Modular Conversational Interaction Framework for Smart Homes, an architecture that aggregates multiple conversational assistants into a cohesive and extensible system. Implemented and evaluated in a living lab context, the framework enables coordinated domotics control, multimodal kitchen assistance, and weather information services. The primary objective is to demonstrate how modular aggregation overcomes interoperability barriers, enhances user experience, and establishes a robust foundation for smart homes that support daily living and health needs, particularly for older adults in home-based care.

2 Related Work

Multi-Domain Conversational Assistants – Research on conversational agents has progressively moved from single-purpose systems toward multi-domain assistants that can manage heterogeneous tasks and services through a unified dialogue interface. A key milestone in this direction is the Schema-Guided Dialogue dataset, which was introduced to support scalable multi-domain conversational agents and zero-shot generalization to unseen services [20]. Building on this, research has continued to explore dialogue modeling strategies that improve robustness in multi-domain settings, including modular architectures and more flexible dialogue state tracking. Collectively, these works demonstrate that scalable conversational interaction requires not only language understanding, but also structured handling of domain-specific knowledge and uncertainty [21, 12].

Multi-Agent and Multi-Assistant Orchestration – A complementary direction is the study of how multiple conversational assistants can be combined into a single user-facing system. Recent work on multi-assistant conversational AI indicates that users benefit when orchestration is managed through a unified interface, rather than requiring them to

choose between assistants [6]. This highlights the value of a coordination layer that can route, aggregate, and mediate among specialized assistants. The challenge extends beyond response generation to include determining which assistant should handle a request, when clarification is needed, and how to manage fallback behavior across different components.

Conversational Interfaces in Smart Homes and Homecare – Building on the coordination challenges outlined above, conversational interfaces have also become increasingly relevant in smart home and home care environments, where users interact with devices, services, and monitoring functions through voice or text. Recent work has shown that voice assistants and conversational user interfaces can improve accessibility and natural interaction in domestic settings, especially when users need to control multiple services without dealing with complex interfaces [1, 16]. However, much of this work still relies on a single assistant tightly coupled to a specific ecosystem or vendor platform, which limits interoperability and extensibility. For that reason, smart-home conversational interaction remains an active research area, particularly in scenarios where multiple services must coexist within a single interaction layer [2, 11].

Conversational Agents for Health Monitoring and Support – Similarly, in health-related settings, conversational agents have been explored for monitoring, education, support, and patient engagement. A literature review of conversational agents in health care reported growing interest in these systems, while also highlighting fragmentation across use cases, dialogue strategies, and evaluation methods [4]. More recent studies continue to emphasize both the potential and the challenges of integrating conversational agents into health workflows, including issues of trust, usability, and professional and patient adoption [14, 15]. Our own work has explored conversational assistants integrated with smart mirrors for mental health assessment [22, 5], alongside the development of a kitchen assistant aimed at supporting active ageing at home, promoting independence through assistance with everyday tasks such as meal preparation and resource management.

Overall, the literature provides strong foundations in multi-domain dialogue modeling, multi-agent coordination, smart-home conversational interaction, and health-supportive conversational agents. Nevertheless, most existing systems remain specialized: they either focus on one assistant, one domain, or one service ecosystem. To address this gap, this work proposes an orchestration layer that integrates multiple specialized conversational assistants, providing a unified interface for smart environments while maintaining back-end modularity. By combining routing, dispatch, and execution mechanisms, users can interact with different assistants without needing to know which service or domain is responsible for each request.

3 Proposed Solution – A Delegation Approach

The proposed framework adopts a delegation-based architecture in which a central conversational agent coordinates user requests and delegates them to specialized domain assistants. This design reflects the Butler like role of the central Assistant, acting as the main entry point of the system, responsible for interpreting high-level intents, maintaining dialogue flow, and forwarding requests to the appropriate component.

This framework is inspired by the organizational structure of traditional manor houses, a model often associated with the upstairs-downstairs hierarchy. In this metaphor, the **butler** acts as the primary interface between the user and the system, overseeing all interactions and managing the distribution of tasks across the available assistants. As Mr. Carson states: *«I am the butler. This is the house I run. Upstairs and down, it is my responsibility»*¹.

¹ *Downton Abbey Script Book Season 1* (HarperCollins, 2012), p. 45.

Supporting the butler are **servant** assistants, each specialized in a specific smart home domain such as domotics control, kitchen assistance, and weather information services, within a Portuguese-language conversational interaction environment.. These assistants operate through a common routing structure that enables systematic task delegation and consistent response handling.

A key advantage of this approach is modularity: new assistants can be introduced or existing ones improved independently, without affecting the central conversational pipeline. The routing layer supports this extensibility by providing structured decisions, confidence scores, and ranked candidate assistants, which allows the decision process to select the most appropriate handler for each request.

Overall, this delegation approach explores coordination, specialization, and separation of concerns. By separating user-facing interaction from domain-specific processing, the framework contributes to a unified conversational experience while preserving flexibility and scalability for future smart home extensions.

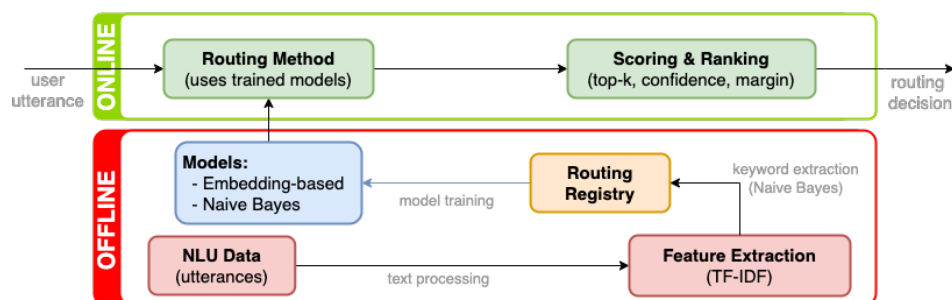
4 Architecture and Processing Pipeline

This section presents the Butler Assistant as a pipeline-oriented architecture consisting of 4 processing stages (Input Reception → Routing → Decision Process → Dispatch).

4.1 Stage 1: Input Reception

The first stage defines the entry point of the Butler Assistant, where user utterances are received. Inputs are assumed to be preprocessed by upstream modalities (e.g., Automatic Speech Recognition) and are therefore provided as textual sequences. Each utterance is assigned a unique `request_id`, enabling end-to-end traceability throughout the pipeline. This identifier allows tracking the processing flow across components and collecting metrics for analysis and evaluation.

4.2 Stage 2: Routing



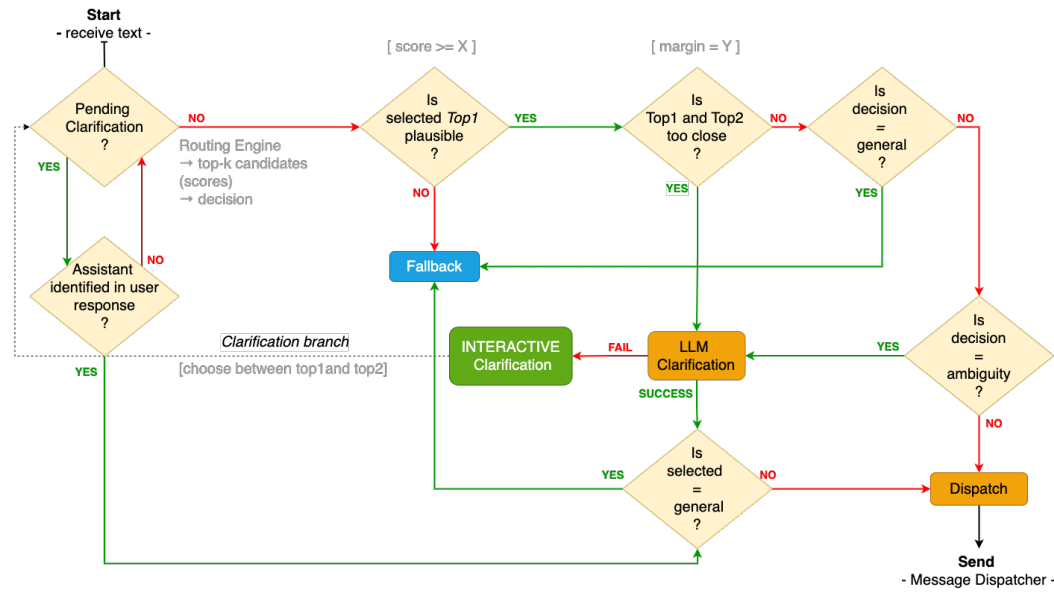
■ **Figure 1** Overview of the routing process, distinguishing between the offline and online phases.

The routing stage is the central linguistic component of the Butler Assistant. Its goal is to determine which domain-specific assistant is the most appropriate handler for a given user utterance by comparing the input against a structured registry of examples, thresholds, and rule-based signals. Rather than returning a simple label, the routing layer produces a structured result containing the selected route, a confidence score, and a ranked list of candidate assistants.

As shown in Figure 1, the routing process is organized into two distinct phases:

- **Offline Phase** – This phase is executed prior to deployment and is responsible for constructing the Routing Registry and training the routing models.
- **Online Phase** – At runtime, incoming user utterances are processed by a routing engine, selected at system initialization, which generates a routing decision based on selected engine.

4.3 Stage 3: Decision Process



■ **Figure 2** Decision flow implemented. For this proposal, $X = 0.25$ represents the minimum plausibility score required for delegation, while $Y = 0.08$ corresponds to the ambiguity margin between candidate assistants..

The **Decision Process**, illustrated in Figure 2, defines how routing outputs are transformed into system actions. Rather than performing linguistic classification, this module evaluates routing confidence and applies system-level control rules to determine the next interaction step. This next step corresponds to a small set of observable outcomes that define the system behaviour at runtime, as summarized in Table 1.

■ **Table 1** Decision Process decision outcomes and their interpretation.

Outcome	Description
dispatch	Routed to a domain-specific assistant
fallback	Handled by the general assistant due to (low confidence/no match)
clarify	System requests disambiguation due to competing candidates
no_prediction	No decision could be produced due to system or processing failure

These outcomes provide a direct mapping between routing signals and system behaviour, and are later used in the evaluation phase to analyse both routing correctness and overall pipeline performance.

4.4 Stage 4: Dispatch

After the **Decision Process** determines the final action, the request is forwarded to the **Message Dispatcher**. This component delivers the utterance to the selected domain-specific assistant and returns the response through the same communication path.

5 Implementation

This section details the implementation of the routing architecture, describing how knowledge is structured, how routing decisions are computed, and how the system resolves uncertainty during interaction. The goal is to provide a clear view of the components that enable efficient, reliable, and context-aware assistant selection.

5.1 Routing Registry

The **Routing Registry** is the central knowledge base of the routing system. It is automatically generated from RASA Natural Language Unit (NLU) files and contains the available routes (the assistants), their representative examples, confidence thresholds, and rule-based signals. The registry also defines global configuration values (Table 2) shared across methods, such as the default threshold, ambiguity margin, top-k value, and fallback route. Because the registry acts as the source of knowledge for the routing process, it defines the available routes, examples, and configuration parameters that guide routing behaviour, while the models determine how routing decisions are computed based on this information.

■ **Table 2** Configurable parameters controlling routing behavior.

Parameter	Value	Description
Global threshold	0.6	Minimum confidence required for a valid match
Ambiguity margin	0.08	Minimum score difference between top candidates
Top-K candidates	3	Number of candidate routes returned by the router
Boosting weight	0.1	Weight applied to rule-based keyword boosting

In addition to example-based representations, the registry incorporates keyword-based rules generated through a statistical feature selection approach. By transforming training utterances into a Term Frequency-Inverse Document Frequency (TF-IDF) representation and applying a Multinomial Naive Bayes model, the system identifies highly discriminative terms for each route. These rules complement the underlying routing models by reinforcing strong lexical signals.

This design is driven by the constraints of real-world conversational systems, where routing must be both efficient and responsive to maintain a fluid interaction. In addition, domain-specific assistants typically provide only a limited number of training examples, making it impractical to rely solely on data-driven approaches. As a result, lightweight models are complemented with rule-based signals, enabling the system to incorporate discriminative cues without increasing training complexity.

5.2 Routing Methods

The **Routing Method** implement a common interface and transforms user input into a standardized routing decision, expressed through a small set of possible outcomes (Table 3). This abstraction allows the system to support different linguistic approaches while preserving

the same output format and downstream behavior. In the current architecture, the Butler Assistant supports two engines: a probabilistic Naive Bayes model and a semantic similarity model based on sentence embeddings.

■ **Table 3** Routing decision outcomes produced by the routing method.

Decision	Description
<code>match</code>	A route exceeds the confidence threshold and is selected
<code>ambiguous</code>	Multiple routes are plausible with insufficient margin
<code>no_match</code>	No route meets the minimum confidence threshold

In addition to model-based scoring, both routing methods incorporate a rule-based boosting mechanism, derived from the routing registry. This mechanism reinforces highly discriminative lexical signals, complementing the statistical or semantic representation used by each model.

The training process follows the routing registry configuration, which is intentionally imbalanced: `domotic` (144 examples), `kitchen` (55), and `weather` (15).

5.2.1 Naive Bayes

The Naive Bayes (NB) Model formulates the routing task as a probabilistic text classification problem, where a user utterance x is assigned to a route $r \in \mathcal{R}$ by maximizing the posterior probability:

$$r^* = \arg \max_{r \in \mathcal{R}} P(r \mid x)$$

Under the Multinomial Naive Bayes assumption, this posterior is computed as:

$$P(r \mid x) \propto P(r) \prod_{i=1}^n P(w_i \mid r)$$

where w_i are the tokens extracted from the utterance x . Utterances are represented using a TF-IDF encoding, which captures term importance across routes based on the registry.

At inference time, the model computes a probability distribution over routes, interpreted as base scores. To enhance discrimination, a rule-based boosting mechanism $b(r \mid x)$ is applied. This component assigns additional weight to routes whose associated keywords (derived from the registry) are present in the input utterance, reinforcing strong lexical cues that may not be fully captured by the probabilistic model alone. The influence of this mechanism is controlled by a global parameter (Table 2).

$$s(r \mid x) = s_{\text{base}}(r \mid x) + b(r \mid x)$$

Candidate routes are ranked, and decisions (Table 3) are made based on the global threshold θ and the ambiguity margin δ .

5.2.2 Embedding-based

The Embedding-based Model implements a semantic similarity problem, representing both user utterances and training examples as dense vectors in a shared embedding space.

Let $\phi(x) \in \mathbb{R}^d$ denote the embedding of an utterance x , computed using a pre-trained sentence encoder. During the offline phase, all example utterances associated with each route r are encoded to form a set of embeddings $\{\phi(x_j^r)\}$.

At inference time, an utterance x is encoded, and its similarity to each route is computed using cosine similarity:

$$\text{sim}(x, x_j^r) = \frac{\phi(x) \cdot \phi(x_j^r)}{\|\phi(x)\| \|\phi(x_j^r)\|}$$

The base score for a route is defined as the maximum similarity across its examples:

$$s_{\text{sem}}(r \mid x) = \max_j \text{sim}(x, x_j^r)$$

This nearest-neighbour strategy captures semantic proximity, enabling the system to generalize beyond surface-level lexical variations and correctly handle paraphrases.

As in the lexical model, a rule-based boosting mechanism $b(r \mid x)$ is applied to reinforce discriminative lexical cues derived from the routing registry. This yields a final score:

$$s(r \mid x) = s_{\text{sem}}(r \mid x) + b(r \mid x)$$

Candidate routes are ranked according to their final scores, and the same decisions (Table 3) are made based on the global threshold θ and the ambiguity margin δ .

5.3 Decision Process

The decision process can be interpreted as a set of conditional steps, summarized as follows:

Pending Clarification – If the system is in a clarification state, the incoming user input is interpreted as a response to a previous disambiguation request. In this case, the decision process directly resolves the decision based on the clarified intent, bypassing standard routing evaluation.

Confidence Check – If no clarification is pending, the decision process evaluates whether the top-ranked candidate meets the minimum confidence threshold (**X** in Figure 2). If no route satisfies this condition, the request is redirected to the fallback mechanism, typically handled by the general assistant.

Decision Interpretation – Following routing, the decision process interprets the predicted decision type. If the decision corresponds to a fallback-related outcome (e.g., **general**), the request is redirected accordingly. If the decision indicates ambiguity, the clarification strategy is triggered.

Ambiguity Detection – When the top candidate exceeds the threshold but the score difference between the top candidates is below the ambiguity margin (**Y**), the input is considered ambiguous. This distinction separates ambiguity detection at the routing level from decision-making at the system level.

Clarification Strategy – In ambiguous cases, the decision process attempts automatic clarification using contextual information. If ambiguity persists, the system enters an interactive clarification state, prompting the user to disambiguate the request. The next user input is then treated as a clarification response.

Final Decision – Once a decision is resolved, the decision process produces a structured `next_step` object that determines how the system proceeds (Listing 1). If the resolved decision corresponds to the general route, fallback handling is applied; otherwise, the request is dispatched to the selected assistant.

■ **Listing 1** Example of a `next_step` object, for *utterance* = Can you turn off all the house lights?

```
"next_step" : {
  "action": "dispatch",
  "assistant": "domotic",
  "message": {
    "type": "dispatch", "assistant": "domotic",
    "text": "Podes desligar todas as luzes da casa?"},
  "meta": {
    "reason": "top_candidate_accepted",
    "confidence": 0.8534,
    "top_k": [
      0: { "route_id": "domotic",
          "score": 0.8534,
          "base_score": 0.7534,
          "boost_score": 0.1},
      1: (...),
      3: { "route_id": "weather",
          "score": 0.0239,
          "base_score": 0.0239,
          "boost_score": 0} ]},
  "request_id": "req_1e687e73dfe4"}
```

6 Evaluation Process

Two experimental scenarios were considered to assess routing performance under different settings:

User input limited to the topics covered by servant assistants (In-domain): Considers utterances drawn exclusively from the domains covered by the system servant assistants (kitchen, domotic, and weather).

Inclusion of user input not covered by servant assistants (Out-of-domain): This setting extends the first by considering also utterances that do not belong to any of the domains supported by servant assistants. These examples simulate unsupported or unexpected user inputs.

The test sets used in both scenarios were constructed from NLU examples that were not used during model training, ensuring a clear separation between training and evaluation data. For the out-of-domain scenario, additional utterances were obtained from the NLU data of an external assistant covering a different domain.

All utterances were processed through the full Butler Assistant pipeline **with Interactive clarification disabled**, ensuring consistent batch evaluation. Each request includes an `expected_route`, enabling supervised evaluation. An observability layer was used to record the outputs of each pipeline stage and capture detailed timing information. This enables fine-grained analysis of system behaviour, including routing decisions, pipeline execution, and end-to-end latency.

7 Results

This section details the experimental results across two distinct scenarios. We begin by evaluating routing performance under both in-domain and out-of-domain conditions, followed by a comparative analysis of system latency.

7.1 Routing Performance: In-domain

The performance (accuracy) of the two routing methods in the **In-Domain** scenario, evaluated at both the routing stage and the final pipeline stage is presented in Figure 3.

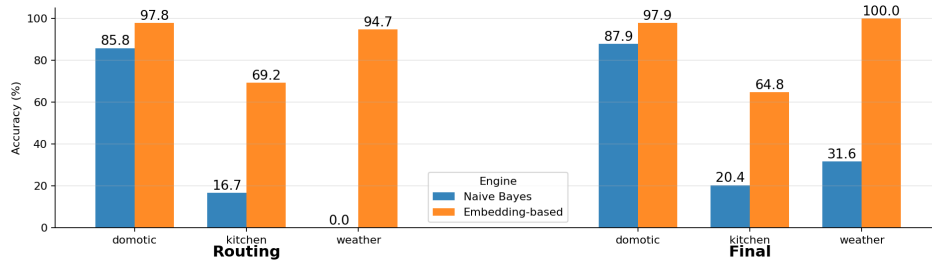


Figure 3 Accuracy per expected route for both routing output and final pipeline output in the **In-Domain** scenario.

A clear and consistent difference between the two routing methods can be observed and the embedding-based model significantly outperforms the Naive Bayes model across all domains. For the embedding-based model, accuracy remains consistently high in both stages, with values close to 100% in the **domotic** and **weather** domains, and above 65% in **kitchen**. The differences between routing and final accuracy are minimal, indicating that correct routing decisions are preserved throughout the pipeline and that downstream components introduce negligible degradation. In contrast, the Naive Bayes model exhibits substantial variability across domains. While performance is relatively strong in **domotic** (approximately 86–88%), it degrades significantly in **kitchen** (around 17–20%) and **weather** (0–32%). This behaviour highlights a strong dependence on training data distribution, with performance decreasing sharply in low-resource domains. Additionally, the gap between routing and final accuracy is limited for the Naive Bayes model, suggesting that most errors originate at the routing stage and are not corrected by subsequent components in the pipeline.

To complement the accuracy metric, confusion matrices were also calculated and analyzed (Figure 4). To ensure the analysis remains independent of dataset imbalance, values are normalized as percentages per expected route, enabling a direct comparison across domains with different sample sizes.

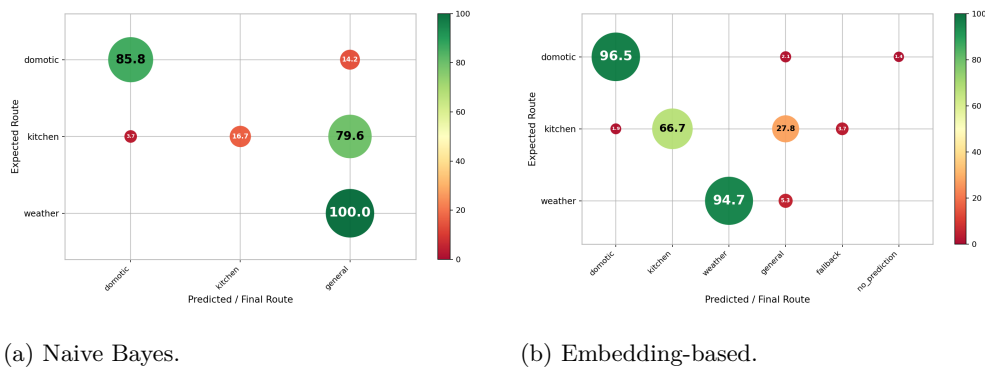


Figure 4 Graphical representation of the confusion matrices for the two routing methods in the in-domain scenario. Values are row-normalized.

The routing behaviour varies significantly across domains, particularly for the Naive Bayes model, while remaining relatively consistent for the embedding-based approach.

For the Naive Bayes model, performance is highly dependent on the target domain. The best results are observed in the **domotic** domain, where most utterances are correctly routed (approximately 85%), reflecting the higher number of training examples. In contrast, performance degrades considerably in **kitchen**, where only a small fraction of utterances is correctly classified, and a large proportion is incorrectly mapped to the **general** route. This effect becomes even more pronounced in the **weather** domain, where the model fails to assign any utterance to the correct route, effectively treating all inputs as **general**. This indicates that, under low-resource conditions, the Naive Bayes model is unable to capture sufficient lexical evidence and defaults to fallback behaviour. In contrast, the embedding-based model exhibits consistently strong performance across all domains. High accuracy is achieved not only in **domotic**, but also in **kitchen** and **weather**, despite the significant differences in training data availability. Errors are relatively limited and more evenly distributed, with only a small proportion of utterances mapped to the **general** route. This suggests that semantic representations allow the model to generalize beyond surface-level lexical patterns, maintaining reliable routing behaviour even in low-resource domains.

7.2 Routing Performance: Out-of-Domain

The performance (accuracy) for the two routing methods on the **Out-Domain** scenario at the various stages is presented in Figure 5.

In this setting, accuracy is computed by considering utterances that do not belong to any supported domain. Each request is associated with an expected out-of-domain label, and correct behaviour corresponds to safely redirecting the input to fallback-related outcomes (**general**, **fallback**, **no_prediction**), rather than assigning it to a domain-specific assistant.

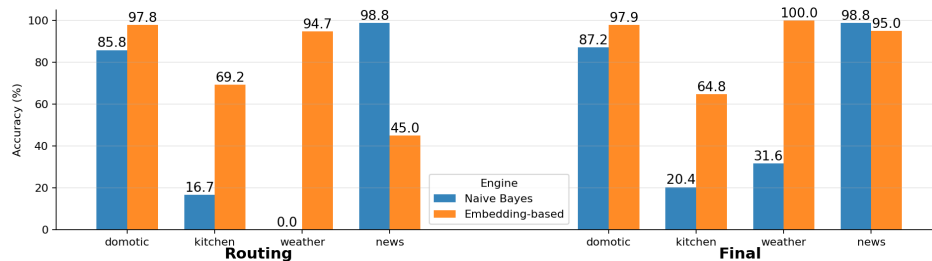


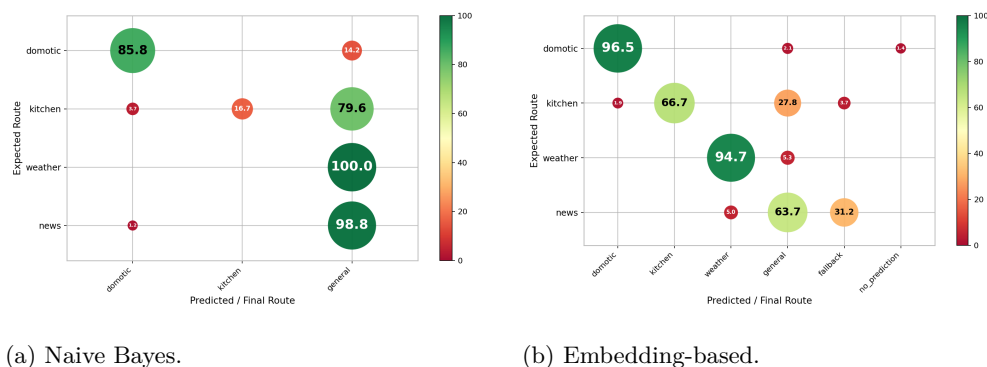
Figure 5 Accuracy per expected route for both routing output and final pipeline output in the **Out-Domain** scenario with unsupported domain (**news**).

Compared with the **In-Domain** setting, this scenario reveals a more differentiated behaviour across routes and across stages of the pipeline. In particular, the main question is not only whether the methods preserve good performance in supported domains, but also whether unsupported inputs (**news**) are correctly handled by the decision process.

For the Naive Bayes model, performance remains highly uneven across routes. The best result is obtained for **news**, where accuracy is already near-perfect at the routing stage and remains so at the final stage. In contrast, the weakest result is observed for **weather**, where routing accuracy drops to zero and only partially recovers in the final output. The **kitchen** route also remains low, while **domotic** preserves comparatively strong performance. Overall, this confirms that the Naive Bayes method remains strongly dependent on route-specific

lexical evidence, producing highly variable results across assistants. For the embedding-based model, routing accuracy remains strong for the supported domains, with very high values in **domotic** and **weather**, and clearly better performance than Naive Bayes in **kitchen**. However, the most interesting behaviour is observed for **news**: while routing accuracy is substantially lower at the routing stage, it increases sharply at the final stage. This shows that, in the **Out-Domain** setting, the decision process plays an important corrective role for the embedding-based method, helping transform uncertain routing outputs into safer final decisions. This contrasts with the **In-Domain** scenario, where the difference between routing and final accuracy was generally small for both methods. Here, the downstream decision logic has a more visible impact, particularly for the embedding-based approach in the unsupported **news** case.

To complement the accuracy analysis, row-normalized confusion matrices were obtained for the out-of-domain scenario, as shown in Figure 6. These matrices provide a detailed view of how unsupported inputs are distributed across predicted routes.



■ **Figure 6** Graphical representation of the confusion matrices for the two routing methods in the out-of-domain scenario. Values are row-normalized.

The analysis focuses on the **news** category, which represents the out-of-domain inputs. In this context, correct behaviour corresponds to mapping these utterances to fallback-related outcomes (**general**, **fallback**, **no_prediction**), rather than dispatching them to a domain-specific assistant.

For the Naive Bayes model, the behaviour is highly conservative. Unsupported **news** utterances are almost entirely mapped to **general**, with negligible assignment to other domains. This confirms that the model rarely attempts semantic generalization and instead defaults to safe rejection when lexical evidence is insufficient. As a result, it avoids incorrect routing to domain-specific assistants, but at the cost of limited flexibility. In contrast, the embedding-based model exhibits a more distributed behaviour. While the majority of **news** utterances are correctly mapped to **general**, a small proportion is assigned to semantically related domains, such as **weather**. This reflects the model's ability to capture semantic similarity, but also highlights a tendency toward over-generalization when handling unsupported inputs.

More importantly, these observations align with the stage-level accuracy results. Although the embedding-based model may initially produce ambiguous or incorrect routing decisions for out-of-domain inputs, the decision process mitigates these effects, increasing the proportion of correct final outcomes. Conversely, the Naive Bayes model shows minimal change between routing and final stages, indicating that its behaviour is largely determined at the routing level. This indicates that the system is able to correctly reject unsupported inputs, avoiding incorrect dispatch to domain-specific assistants.

7.3 Latency

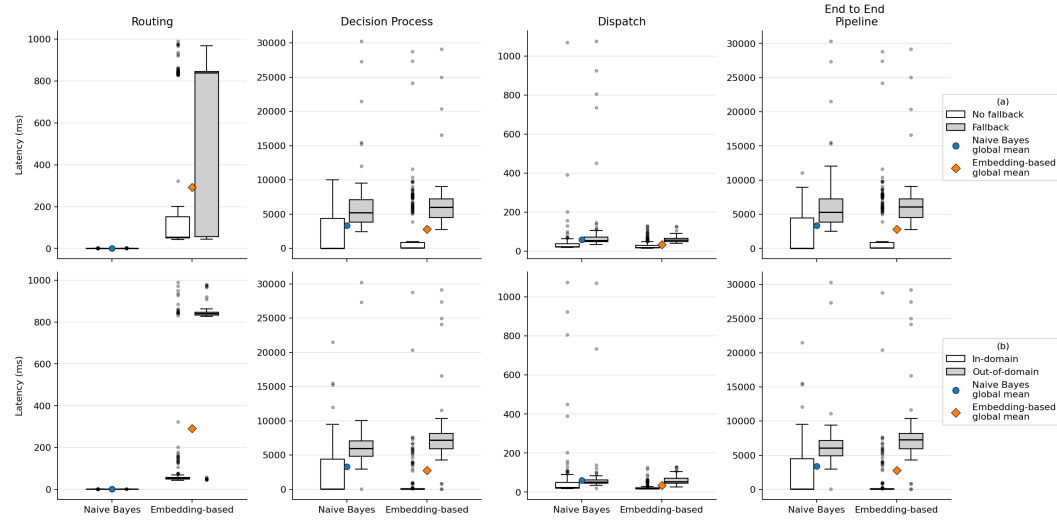


Figure 7 Latency for the pipeline stages. Top row presents results as function of routing method (x-axis) and activation or not of fallback during decision process. Bottom row presents results as function of the evaluation scenario (*In-Domain* vs *Out-of-Domain*). Boxplots are complemented with markers to ensure legibility of the visualizations, different y-axis ranges were applied to the individual stages..

An examination of both mean and median latencies shows that the system’s stages generally operate within time frames compatible with a usable conversational assistant, responding quickly enough to sustain natural dialog between users and the assistant. Across all configurations, median latency remains low for most components, particularly in the **Routing** and **Dispatch** stages, confirming that the core pipeline consistently meets the responsiveness requirements of real-time interaction.

The latency analysis further reveals a clear distinction between typical system behaviour and occasional high-cost interactions. While most interactions are processed quickly, a small number of cases exhibit significantly higher latency. When comparing interactions that involve fallback with those that do not (shown in the top row of Figure 7) it becomes evident that high-latency cases occur exclusively under the **Fallback** condition. When examining the bottom row of the plot, which separates in-domain and out-of-domain scenarios, similar high-latency cases appear in both conditions. The same pattern appears under both routing approaches (**Naive Bayes** and **Embedding-based**), confirming that extreme latency values arise from the fallback mechanism rather than from the choice of routing strategy.

The results make clear that the **Decision Process** stage is the primary contributor to end-to-end latency. Whereas routing and dispatch remain reliably fast, the decision component introduces significant fluctuations, most notably in fallback situations. These fluctuations propagate downstream, causing the overall latency to depend chiefly on the cost of decision-making under uncertainty. The presence of these high-latency cases can therefore be directly attributed to the fallback mechanism, which relies on external LLM-based processing. Unlike the rest of the pipeline, this component is less controllable because it depends on external conditions and resources such as model inference time, network latency, and variability in upstream services. These dependencies explain why fallback interactions can lead to occasional but substantially slower responses compared with typical system behaviour.

These results show that responsiveness is limited by uncertain inputs, not routing. Reducing fallbacks and streamlining downstream decisions (especially those using external services) is key to lowering latency.

8 Conclusion and Future Work

The challenge of providing a unified conversational interface over domain-specific assistants was addressed by ensuring that routing decisions remain accurate, robust, and safe under both in-domain and out-of-domain conditions. To tackle this problem, a modular routing architecture was proposed in which user interaction is decoupled from underlying services through a structured decision layer, where model-based routing is combined with explicit system-level policies for ambiguity handling and fallback control.

Superior performance was achieved by the embedding-based approach, especially in low-resource domains where its ability to generalize beyond lexical similarity proved advantageous. Furthermore, both routing methods exhibited stable behaviour in out-of-domain conditions, consistently avoiding mis-dispatches through fallback activation.

In terms of latency, the system maintains response times appropriate for real-time use, with end-to-end latency shaped mainly by pipeline-level operations rather than by the routing methods.

Future Work

While the results are encouraging, a number of limitations still remain. Naïve Bayes is sensitive to vocabulary coverage, while semantic models may introduce ambiguity through similarity-based matching. These results stress the importance of explicit uncertainty-handling mechanisms, such as ambiguity detection and fallback, to maintain reliable behaviour in multi-domain systems. Aligned with these results, a number of promising avenues for further development become apparent:

- **Improvement of Fallback** – Exploring faster and less network dependent alternatives.
- **Hybrid Fusion Model** – Develop learned fusion strategies that dynamically combine lexical and semantic information, enabling the system to balance robustness and generalization depending on the input characteristics.
- **Adaptive decision** – Extend the decision layer to incorporate adaptive mechanisms that adjust routing thresholds and ambiguity margins based on observed system behaviour. This includes dynamically tuning confidence thresholds and decision boundaries according to performance metrics, enabling the system to better handle variability across domains and improve robustness over time.
- **Decomposition of Complex Commands** – Extend the system to handle more complex user utterances by incorporating techniques such as chunking and/or syntactic/semantic parsing. This will enable the system to decompose a single input into multiple actionable commands and route each one to the appropriate servant assistant.

References

- 1 Saul Albert, Magnus Hamann, and Elizabeth Stokoe. Conversational user interfaces in smart homecare interactions: A conversation analytic case study. In *Proceedings of the 5th International Conference on Conversational User Interfaces, CUI '23*, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3571884.3597140.

- 2 Anastasios Alexiadis, Alexandros Nizamis, Ioannis Koskinas, Dimosthenis Ioannidis, Konstantinos Votis, and Dimitrios Tzovaras. Applying an Intelligent Personal Agent on a Smart Home Using a Novel Dialogue Generator. In Ilias Maglogiannis, Lazaros Iliadis, and Elias Pimenidis, editors, *Artificial Intelligence Applications and Innovations*, pages 384–395, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-49186-4_32.
- 3 Frank Bentley, Chris Luvogt, Max Silverman, Rushani Wirasinghe, Brooke White, and Danielle Lottridge. Understanding the Long-Term Use of Smart Speaker Assistants. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(3):91:1–91:24, 2018. doi:10.1145/3264901.
- 4 Lorainne Tudor Car, Dhakshenya Ardhithy Dhinakaran, Bhone Myint Kyaw, Tobias Kowatsch, Shafiq Joty, Yin-Leng Theng, and Rifat Atun. Conversational Agents in Health Care: Scoping Review and Conceptual Analysis. *Journal of Medical Internet Research*, 22(8):e17158, 2020. doi:10.2196/17158.
- 5 Pedro Carneiro, Inês Águia, David Palricas, Ana Patrícia Rocha, António Teixeira, and Nuno Almeida. Kitchen Assistant for Active Ageing at Home. In *Proceedings of the 11th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion*, DSAI '24, pages 193–202, New York, NY, USA, 2025. Association for Computing Machinery. doi:10.1145/3696593.3696624.
- 6 Christopher Clarke, Joseph Peper, Karthik Krishnamurthy, Walter Talamonti, Kevin Leach, Walter Lasecki, Yiping Kang, Lingjia Tang, and Jason Mars. One Agent To Rule Them All: Towards Multi-agent Conversational AI. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3258–3267, Dublin, Ireland, 2022. Association for Computational Linguistics. doi:10.18653/v1/2022.findings-acl.257.
- 7 European Commission. Final report on consumer Internet of Things sector inquiry. URL: https://ec.europa.eu/commission/presscorner/detail/en/ip_22_402.
- 8 Colin Crawford. Protocol power: Matter, IoT interoperability, and a critique of industry self-regulation, 2024. doi:10.14763/2024.2.1776.
- 9 Inge Dhamanti, Ika M. Nia, Krishnaraj Nagappan, and Balaji P. Srikanth. Smart home healthcare for chronic disease management: A scoping review. *DIGITAL HEALTH*, 9:20552076231218144, 2023. doi:10.1177/20552076231218144.
- 10 Anh Duong and Maria Valero. Usability of Voice Assistants in Healthcare: A Systematic Literature Review. In Dario Salvi, Pieter Van Gorp, and Syed Ahmar Shah, editors, *Pervasive Computing Technologies for Healthcare*, pages 386–401, Cham, 2024. Springer Nature Switzerland. doi:10.1007/978-3-031-59717-6_25.
- 11 Maksym Ketsmur, António Teixeira, Nuno Almeida, and Samuel Silva. Towards European Portuguese Conversational Assistants for Smart Homes. In Ricardo Rodrigues, Jan Janoušek, Luís Ferreira, Luísa Coheur, Fernando Batista, and Hugo Gonçalo Oliveira, editors, *8th Symposium on Languages, Applications and Technologies (SLATE 2019)*, volume 74 of *Open Access Series in Informatics (OASICS)*, pages 5:1–5:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/OASICS.SLATE.2019.5.
- 12 Lizi Liao, Tongyao Zhu, Le Hong Long, and Tat Seng Chua. Multi-domain Dialogue State Tracking with Recursive Inference. In *Proceedings of the Web Conference 2021*, WWW '21, pages 2568–2577, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3442381.3450134.
- 13 Mingzhou Liu, Caixia Wang, and Jing Hu. Older adults; intention to use voice assistants: Usability and emotional needs. *Heliyon*, 9, November 2023. doi: 10.1016/j.heliyon.2023.e21932. doi:10.1016/j.heliyon.2023.e21932.
- 14 A. Luke MacNeill, Lillian MacNeill, Alison Luke, and Shelley Doucet. Health Professionals' Views on the Use of Conversational Agents for Health Care: Qualitative Descriptive Study. *Journal of Medical Internet Research*, 26(1):e49387, 2024. doi:10.2196/49387.
- 15 Laura Martinengo, Xiaowen Lin, Ahmad Ishqi Jabir, Tobias Kowatsch, Rifat Atun, Josip Car, and Lorainne Tudor Car. Conversational Agents in Health Care: Expert Interviews to Inform the Definition, Classification, and Conceptual Framework. *Journal of Medical Internet Research*, 25(1):e50767, 2023. doi:10.2196/50767.

- 16 Bettina Minder, Patricia Wolf, Matthias Baldauf, and Surabhi Verma. Voice assistants in private households: a conceptual framework for future research in an interdisciplinary field. *Humanities and Social Sciences Communications*, 10(1):173, 2023. doi:10.1057/s41599-023-01615-z.
- 17 Martin Porcheron, Joel E. Fischer, Stuart Reeves, and Sarah Sharples. Voice interfaces in everyday life. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 1–12, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3173574.3174214.
- 18 Alisha Pradhan, Kanika Mehta, and Leah Findlater. “Accessibility Came by Accident”: Use of Voice-Controlled Intelligent Personal Assistants by People with Disabilities. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 1–13, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3173574.3174033.
- 19 Parisa Rashidi and Alex Mihailidis. A Survey on Ambient-Assisted Living Tools for Older Adults. *IEEE Journal of Biomedical and Health Informatics*, 17(3):579–590, 2013. doi:10.1109/JBHI.2012.2234129.
- 20 Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8689–8696, April 2020. doi:10.1609/aaai.v34i05.6394.
- 21 Dingmin Wang, Chenghua Lin, Qi Liu, and Kam-Fai Wong. Fast and Scalable Dialogue State Tracking with Explicit Modular Decomposition. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–295, Online, 2021. Association for Computational Linguistics. doi:10.18653/v1/2021.naacl-main.27.
- 22 Inês Águia, Pedro Carneiro, Rafael Pinto, Ana Patrícia Rocha, Nuno Almeida, and António Teixeira. Integrating Conversational Assistants with Smart Mirrors for Mental Health Assessment. In *Companion of the 2025 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp Companion '25, pages 1202–1208, New York, NY, USA, 2026. Association for Computing Machinery. doi:10.1145/3714394.3756237.

From Natural Language to Structured Queries

An LLM-Powered Chatbot for a Smart Tourism Observatory

Rodrigo Bravo Simões¹ ✉ 

Instituto Universitário de Lisboa (Iscte-IUL), Portugal

Adriano Lopes ✉ 

Instituto Universitário de Lisboa (Iscte-IUL), Portugal

Fernando Brito e Abreu ✉ 

Instituto Universitário de Lisboa (Iscte-IUL), Portugal

Abstract

The accessibility of Smart Tourism Initiatives and Tools (STITs) is a cornerstone of digital transformation in the tourism sector. The EUROSTIT project addresses this through a European Observatory that catalogs these technologies. However, sifting through hundreds of results remains a significant challenge for users. This paper presents a specialized conversational agent that facilitates discovery through a filter-based natural language interface. Departing from computationally intensive fine-tuning, our architecture uses prompt engineering – specifically, one-shot Chain-of-Thought (CoT) reasoning – to generate boolean search queries (BSQs). This approach enables the system to distinguish conversational fillers and other irrelevant terms from domain-specific smart tourism entities, autonomously translating free-text into structured database queries. We describe a dynamic web application with a synchronized user experience where conversational input acts as cumulative filters. The performance of different LLMs is compared through various metrics. We conducted quantitative evaluations to gain insight into the feasibility of using readily available commercial LLMs, instead of resource-heavy model fine-tuning for domain-specific boolean search query generation.

2012 ACM Subject Classification Computing methodologies → Natural language processing; Human-centered computing → Natural language interfaces; Information systems → Information retrieval query processing; Information systems → Search interfaces

Keywords and phrases Large Language Models (LLMs), Prompt Engineering, Chain-of-Thought (CoT), Boolean Search Query (BSQ) Generation, Natural Language Interfaces (NLI), Smart Tourism, Chatbots

Digital Object Identifier 10.4230/OASICS.SLATE.2026.17

Supplementary Material

Software (Source Code): <https://github.com/EUROSTIT/Intelligent-Interaction>

Dataset: <https://zenodo.org/records/19742395>

Funding This work was partly supported by project 2024.07579.IACDC/2024, funded by FCT – Fundação para a Ciência e a Tecnologia, I.P., under PRR – Plano de Recuperação e Resiliência funding, investment RE-C05-i08 – Ciência Mais Digital and, partially supported by national funds through FCT, ISTAR-Iscte Pluriannual Projects UID/04466/2025.

1 Introduction

The adoption of digital technologies in the tourism sector depends heavily on the accessibility of Smart Tourism Tools (STTs) and the lessons learned from their implementation in Smart Tourism Initiatives (STIs). Making that information publicly available was the motivation

¹ Corresponding author



underlying EUROSTIT project (European Observatory of Smart Tourism Initiatives and Tools), which successfully developed a digital infrastructure to catalog and classify European smart tourism technologies, available online at <https://www.eurostit.eu/>.

The EUROSTIT content is extracted from several sources, namely PDF catalogs created by tourism-related organizations [10, 15]. Such catalogs are not prone to online search, and their content is not structured or easily indexable. As the number of collected solutions grows, the value of the observatory increasingly depends on the ability to: (i) organize solutions consistently according to an adequate taxonomy [11] and (ii) allow tourism stakeholders to perform searches that retrieve STITs (Smart Tourism Initiatives and Tools) aligning with their declared aim.

The first concern is addressed in [9], and the second concern is addressed in the current paper, where we describe how we improved user interaction and facilitated the discovery of STITs within the observatory by developing an intelligent conversational agent supporting natural language interaction. Each step of the interaction introduces cumulative filters that reduce the universe of relevant STITs. This paper details the requirements, architectural design, implementation, and evaluation of this filter-based conversational agent.

The remainder of this paper is organized as follows. Section 2 reviews related work in the context of conversational agents in tourism. Section 3 details the system requirements and architecture, user interface, and prompt engineering methodology. Section 4 presents the evaluation of the implemented system. Section 5 provides a discussion of methodological choices and research implications, and Section 6 concludes the article with directions for future work.

2 Related Work

The integration of conversational artificial intelligence (AI) and chatbots into the tourism and hospitality sectors has experienced significant growth over the past decade. This surge is driven by the industry's need for personalized, 24/7 customer service and the optimization of digital ecosystems within smart tourism destinations.

Several systematic literature reviews have recently mapped the landscape of conversational agents in this domain. Benaddi et al. [4, 5] provided a comprehensive analysis of chatbot evolution, classifying their development from simple rule-based decision trees to advanced AI-driven systems. Their review highlights how chatbots impact the tourism industry within the “6A framework” [6] (Attractions, Amenities, Accessibility, Ancillary services, Available packages, and Activities), demonstrating their utility in helping users in all phases of the travel journey [4].

The behavioral and psychological dimensions of user interaction are explored by Alharbi et al. [2], who proposed an integrated conceptual framework that details the mechanisms, modifiers, and implications of AI-enabled conversational agents (AICAs). Their systematic review emphasizes the critical importance of emotional, ethical, and trust-related factors in AICA adoption models that influence user engagement [2].

In terms of practical applications, much of the existing research was centered on end-consumer applications, particularly those related to cultural heritage and tourist routing. For example, HeriBot [7] uses latent Dirichlet allocation (LDA) for text analysis and was later extended with ontologies that represent knowledge about destinations [8]. Future work on HeriBot will incorporate deep learning, which has already been employed in another chatbot to support tourist trips using Gated Recurrent Unit (GRU) cells² [16]. These chatbots can

² Basic computational units of a GRU neural network, a type of recurrent neural network (RNN) designed to process sequential data such as text, time series, speech, or sensor streams.

perform tasks such as recommending points of interest and building customized itineraries. Mobile chatbots have also been successfully deployed to help tourists with local navigation and recommendations [3].

More recently, the focus has shifted toward the disruptive potential of Generative AI in smart tourism. Kontogianni and Alepis [13] examined how LLMs (Large Language Models) and, in particular, Generative Pre-trained Transformers (GPTs) are improving tourist experiences and streamlining operations. Saleh [14] investigated AI integration patterns and prompt optimization techniques, highlighting the operational impacts of using models such as ChatGPT, Claude, and Gemini in hospitality settings.

Although the literature extensively covers B2C (Business-to-Consumer) chatbots designed for tourist itinerary planning or customer service, there is a noticeable gap with respect to tools designed to assist stakeholders in navigating complex technological ecosystems, such as repositories of Smart Tourism Initiatives and Tools. Our work departs from these paradigms: instead of directing tourists to physical attractions, the EUROSTIT chatbot is designed to direct tourism operators and researchers to appropriate digital technologies.

Another aspect worth considering is related to the use of LLMs for generating boolean search queries (BSQs)³. Research suggests that fine-tuning LLMs to generate BSQs yields better results than prompt engineering [1]. However, fine-tuning is not available for proprietary models and is computationally expensive. A 2023 study [17] evaluated prompt engineering with ChatGPT for BSQ generation and concluded that this approach had potential, but noted that its results had a lower recall than state-of-the-art alternatives at the time. Since commercial LLMs have improved since then, it is useful to revisit this problem using better models. A 2025 study [12] compared different commercially available LLMs from OpenAI to evaluate the quality of the generated BSQs. However, the methodology rewards models that generate more complex queries with many syntactical elements without analyzing their search results using relevance metrics such as precision and recall. Thus, we identify the comparison of newer LLMs for BSQ generation without fine-tuning as a research gap that we aim to address.

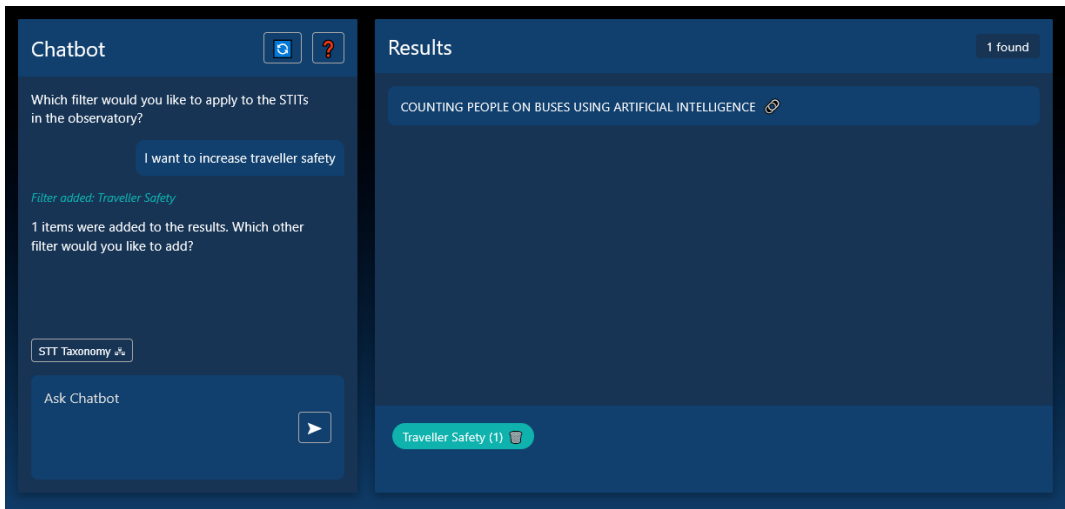
3 Methodology and Architecture

The proposed chatbot architecture follows the classic three-tier model, comprising a presentation tier, a logic tier, and a data tier. The chatbot itself is implemented as a dynamic web application that can be integrated into virtually any content management system (CMS), although some integration effort may be required depending on the target platform. The following subsections describe how the chatbot's components map to the three tiers and discuss the associated implementation challenges.

3.1 User Interface Design

In this presentation tier, the chatbot user interface (UI) is inherently simple, as the primary focus is to facilitate sequential and easy-to-follow conversations. However, the proposed chatbot deviates from traditional free-text conversational interfaces. After discussion with various stakeholders, the functional requirements drawn for the chatbot highlight the need for users to interact primarily via keyboard and to view a continuously updated list of relevant STITs based on applied filters, and also to allow filtering by STT category according to

³ Search expressions that use Boolean operators to combine keywords and control how a search engine, database, or digital library retrieves results.



■ **Figure 1** Chatbot UI, featuring the conversational pane, applied filters, and results list.



■ **Figure 2** The Taxonomy Pane.

the taxonomy presented by Galvão et al. [11]⁴ As for the non-functional requirements, they dictate that the system must be a responsive web application, capable of operating across multiple screen sizes.

The initial user interface mockups explored a three-pane desktop layout comprising a chatbot pane, a filters pane, and a results pane. Following stakeholder discussions, this was optimized into a two-pane layout to conserve screen real estate, where the list of applied filters is shown at the bottom of the results pane. Figure 1 presents the UI.

In the chatbot pane above the input area, there is a button that provides access to a taxonomy pane (see Figure 2). This pane allows users to manually select or deselect categories from the STT taxonomy, effectively adding or removing filters. Additional features include a reset button to clear the conversation and a help button to guide new users, both at the top of the chatbot pane.

⁴ A summary of this taxonomy is available in the observatory's Taxonomy page.

3.2 Application Intelligence and Data Integration

At the core of the application’s intelligence is the ability to transform a user’s natural language input into a boolean search query via an LLM. This query can contain the boolean operators AND and OR, as well as grouping to obtain higher precedence. For instance, given the input “*I am searching for sustainable transport*”, a possible resulting query could be ((Sustainability OR Sustainable) AND (Transport OR Mobility)). The query is then run against the titles and descriptions of STITs in the CMS’s database.

To perform this transformation, the LLM receives a set of instructions that indicate which terms should be ignored and which should be retained as keywords, as shown in Table 1. However, the examples in table 1 are incomplete and insufficient because unexpected types of terms may be present in the user’s input. To handle these unforeseen categories, the LLM is instructed to categorize the terms in the user prompt as relevant or irrelevant. This is an instance of Chain-of-Thought prompting since the model is instructed to generate an intermediate reasoning step.

■ **Table 1** Categorization of User Input Terms for Query Rewriting.

Category	Examples
Irrelevant (Conversational Filler)	“I want”, “show me”
Irrelevant (Goal Verbs/Subjective)	“improve”, “boost”, “best”, “top”
Irrelevant (Generic Domain Terms)	“solution”, “technology”
Relevant (Physical Objects/Concepts)	“bus”, “museum”, “safety”, “sustainability”
Relevant (Specific Technologies)	“5G”, “app”

Additional instructions were included in the system prompt to handle certain edge cases, which we will describe below.

The models are instructed to avoid abbreviations such as “AR” (Augmented Reality) because the database query returns items containing these characters in any position (for example, “AR” would match the word “car”). Pluralization is another issue. The keyword “sensors” would not match text containing “sensor” in the singular, so the singular form is preferred for generated keywords. However, if the plural form changes the spelling of the base word (e.g., “city” vs. “cities”), both the singular and plural forms should be included in a boolean OR group.

The LLM is instructed to respond with JSON containing a recursive representation of the structured query, a short title for the filter, and the relevance classification of the original terms. The application handles cases where the LLM returns an invalid JSON.

We use a one-shot prompting technique. Therefore, the system prompt concludes with an example of user input and a possible response that adheres to the given instructions.

The system uses OpenRouter, a unified interface for LLMs that offloads computation of inference and facilitates model comparison and the implementation of fallback mechanisms. In our chatbot implementation, a configuration file contains an ordered list of models to try. If a model is unavailable or removed from the platform, fallback models can be used. This is also useful if the OpenRouter API key expires or a credit limit is reached, as free models can instead be tried.

Integrating the STT taxonomy into the application involves a simple database lookup based on the user’s selection in the taxonomy pane. This is because the STTs in the observatory were previously annotated with properties that indicate their taxonomic categories..

3.3 Technology Stack

During the technology exploration phase, several existing chatbot libraries and templates were evaluated, including BotUI, react-chatbot-kit, Streamlit, and Gradio. However, these solutions were deemed too limited to support the envisioned filter-based paradigm without substantial modifications. Even promising templates, such as Next.js Chatbot available on Vercel’s website, presented limitations regarding the integration of arbitrary LLMs due to their reliance on specific SDKs. Consequently, a custom frontend development approach was adopted.

The application was built from scratch using the Next.js framework, which combines React for the frontend and Node.js for the backend. The chatbot is decoupled from the observatory’s primary Content Management System, Omeka S, ensuring that it can be more easily integrated into another CMS in the future, if necessary. TypeScript was selected as the programming language to leverage static typing and better avoid runtime errors. Furthermore, to foster transparency, reproducibility and future research within the smart tourism domain, the code produced for this project is open-source at <https://github.com/EUROSTIT>.

As an example of some work required to integrate the chatbot and the CMS, we developed an Omeka S module (extension) to create a table containing the title and description of each item, with a full-text index. This table is automatically synchronized when an item is created, updated, or deleted. The LLM generated boolean search JSON is converted into MySQL syntax for boolean full-text search expressions. This allows for efficient retrieval of matching items while protecting against SQL injection through the use of a prepared statement in the code.

4 Evaluation

4.1 Evaluation Methodology

We evaluated the performance of various large language models (LLMs) on the task of converting a natural language input prompt into a structured query to search the observatory’s database. To achieve this, we conducted quantitative evaluations based on a manually curated set of 18 user prompts of varying complexity.

In addition to this, we analysed the structured queries generated by the model to check whether they matched or contradicted the instructions provided in the system prompt.

For the quantitative evaluation, we collected the set of relevant STITs that should appear in the chatbot’s results for each of the 18 prompts. By comparing this set with the results obtained by running the generated query against the database, we could calculate precision and recall metrics. We also collected response times for model inference, enabling us to analyse the trade-offs between speed, quality and cost.

At the time of writing, the observatory contains more than a thousand STITs. Screening them all manually to collect those relevant to a specific keyword would be too time-consuming. Therefore, we first sort the STITs by relevance to a keyword using embeddings. Specifically, we sorted them according to the distance between the keyword’s embedding and the embedding of a concatenation of the STIT’s title and description. We used a model available on OpenRouter (perplexity/pplx-embed-v1-4b) to determine those embeddings.

For prompts containing multiple relevant keywords, we created short natural language phrases that connected the keywords. Initially, we tried another approach where two keywords, conceptually joined by a boolean AND, were individually turned into embeddings. The list of items was then sorted by the highest distance between the various keyword embeddings

and the item’s embeddings. After testing both approaches on a few prompts, we found that using a single embedding to represent a more complex concept was more appropriate. Once such a list is generated, manual screening can be performed more efficiently because, at a certain point, the items become less relevant. During the screening, if 10 consecutive items are marked as irrelevant, the process ends and all unmarked items are considered irrelevant.

In our evaluation methodology, we compared high- and low-priced models from the four providers with the highest market share in OpenRouter as of April 5, 2026: Qwen, Google, OpenAI, and Anthropic. To ensure a fair comparison, the models were selected so that the higher-priced options shared a similar price point across all providers, with the same principle applied to the lower-priced tier. Models available free of charge were excluded from this evaluation because one provider did not have free models available in OpenRouter, and the free models of two other providers were rate-limited whenever we attempted to conduct our evaluation.

The dataset used for this evaluation is available at <https://zenodo.org/records/19742395>. It includes prompts, results of manual screening, boolean search queries generated by the LLMs, and items matched by full-text search.

4.2 Test Set Characterization

The test set contains manually created sets of relevant items for each of the 18 prompts. On average, there are 31 relevant items per prompt, with a standard deviation of 21, ranging from a minimum of 7 to a maximum of 87 items.

The test set contains six simple prompts, six medium-complexity prompts, and six complex prompts. The level of complexity is related to the number of relevant concepts for query construction: simple prompts have one concept, medium-complexity prompts have two, and complex prompts have more. The suitability of these prompts was assessed during manual screening. Several prompts were discarded because they targeted a very specific niche with little representation in the observatory and had very few relevant items. For simple and medium-complexity prompts, we discarded those with fewer than 10 corresponding STITs. However, it was challenging to find many relevant items for complex prompts, so we accepted those with fewer items. Some of the prompts in the test set are transcribed in the following subsections. The complete list of prompts can be found in the supplementary material.

4.3 Evaluation of Instruction-Following

We will now present some interesting findings regarding boolean queries generated from user prompts in the test set. All of the model responses were syntactically valid JSON and generally adhered to the instructions in the system prompt. The main difference was in the synonyms generated in the OR groups. In the following, we discuss some prompts, referring to them by the position they appear in the published test set.

Prompt #1 is *“Usage of chatbots for tourism”*. When considering that all items in the observatory should be related to tourism, this prompt is simple because it contains only one keyword that is relevant for searching: “chatbot”. As previously mentioned, the Chain-of-Thought step was added to the system prompt to better equip the model to consider terms such as “tourism” as filler. However, only one model has done so (qwen/qwen-max). All other models marked two terms for keeping (“chatbot” and “tourism”), while all models correctly ignored all the other terms in the user prompt.

Prompt #2 is *“I am searching for sustainable transport.”*. Some models identified the concept “sustainable transport” as a single unit, while others separated it into two concepts. These two different approaches to the analysis step lead to two different query structures.

■ **Table 2** Metrics of the evaluation for each model.

Model	Precision	Recall	F1 Score	Latency (ms)
qwen/qwen3-235b-a22b-2507	0.53	0.28	0.37	6305
qwen/qwen-max	0.59	0.28	0.38	2003
google/gemini-2.5-flash	0.52	0.36	0.43	1397
google/gemini-2.5-pro	0.71	0.42	0.53	7965
openai/gpt-4.1-nano	0.57	0.21	0.30	1994
openai/gpt-5.2	0.49	0.20	0.29	6221
anthropic/claude-3-haiku	0.46	0.27	0.34	1800
anthropic/claude-haiku-4.5	0.48	0.33	0.39	1844

In the first case, a single OR grouped synonyms for “sustainable transport”, while in the second case, an AND joined two OR groups, one for each term. These different approaches to grouping terms into keywords can be seen in the results of other prompts in the test set, with similar consequences. The system prompt instructs the models to avoid grouping terms unless they form a well-known phrase. However, it does not prescribe a methodology for deciding when to group terms. Therefore, both approaches are correct.

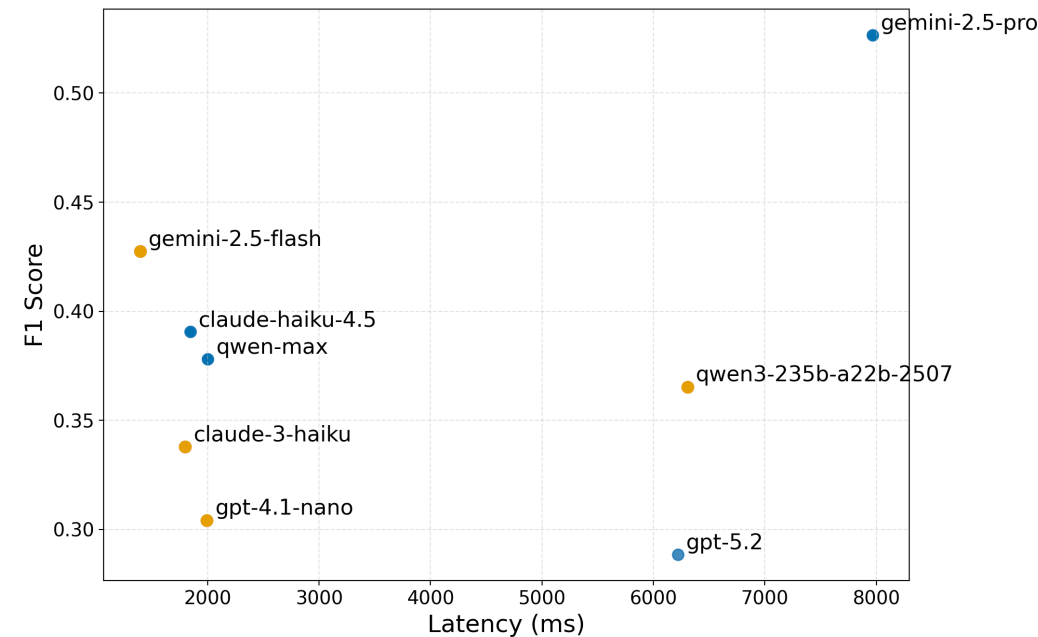
Prompt #5 was designed to have a lot of filler: *“I was thinking that some type of solutions for improving traveller health could be interesting for my business”*. As with most prompts in the test set, most of the evaluated models correctly ignore the filler. An exception for this prompt is anthropic/claude-3-haiku, which does not exclude “solution” and in fact includes the generic domain terms “solution”, “initiative”, and “tool”, as well as the subjective goal terms “improve”, “enhance”, and “boost”. Some of these synonyms were included in the system prompt as examples of what not to include.

4.4 Quantitative Evaluation

The macro results of the quantitative evaluation are shown in Table 2 and Figure 3. Precision, recall, and latency were calculated as the average of those metrics for the prompts in the test set. The F1 score was calculated based on the averages of precision and recall, and it is the most relevant metric as it combines both the relevance of the items matched by the query and the coverage of the relevant items in the observatory.

The model google/gemini-2.5-pro dominates all others in terms of precision and recall. However, it is the slowest, with an average latency of almost eight seconds. If latency is a concern, google/gemini-2.5-flash would be a better alternative, as can be seen in Figure 3. It is both the fastest model and the second best in terms of F1 score. It could be expected that higher-priced models outperform lower-priced models from the same provider, which is true for three of these providers; however, openai/gpt-4.1-nano has a negligibly higher F1 score than openai/gpt-5.2 (0.30 versus 0.29).

Regarding the previously mentioned prompt #1, in which only Qwen’s qwen/qwen-max correctly identified “tourism” as a term to exclude from the search query, it is interesting to note that the lower-priced model from the same provider (qwen/qwen3-235b-a22b-2507) has a higher F1 score for that prompt (0.53 versus 0.38). An analysis of the generated queries reveals that the higher-priced model did not generate synonyms of the keyword “chatbot” (such as “virtual assistant”), while the lower-priced model did. This explains the difference in F1 scores despite the error identified in the previous subsection.



■ **Figure 3** Plot of models’ F1 Score versus Latency. Higher-priced models are marked dark blue, while lower-priced models are light orange.

5 Discussion

A primary methodological decision was to forgo model fine-tuning in favor of one-shot Chain-of-Thought prompting. Although fine-tuning provides domain adaptation, it frequently involves substantial additional data labeling and computing resources. However, the findings of this study indicate that the reliance on prompt engineering of readily available LLMs remains constrained. Our quantitative evaluation indicates that even the model with the highest precision and recall values exhibits a precision of only 0.70, signifying that almost one-third of the items it returns are irrelevant. Furthermore, the recall value is 0.41, indicating that it does not return more than half of the relevant items in the observatory.

The construction of the test set used in this study also merits discussion. A single human annotator was assigned to label items in the observatory as relevant or irrelevant for a given user prompt. The efficacy of this process was improved using embeddings to sort items according to their relevance, as detailed in Subsection 4.1. However, the execution of this labeling task requires expertise that the annotator may not possess. For example, the annotator may not discern that an item mentioning “barrier-free tourism” is pertinent to a prompt inquiry about accessibility in tourism. Another issue that has been identified is the ambiguity that may be present in the user prompt. One of the inquiries posed in the test set pertains to “booking systems,” thereby prompting the following question: do ticketing systems constitute booking systems? The question arises as to whether only systems utilized for the reservation of a particular, time-bound resource should be designated as booking systems. It is important to note that different annotators may interpret this differently, just as different models can generate boolean queries that imply different interpretations.

6 Conclusion and Future Work

This paper presents the design and implementation of an LLM-powered, filter-based conversational agent for the EUROSTIT Observatory. Through prompt engineering, we developed a system that translates natural language into structured boolean search queries, making it easier to find Smart Tourism Initiatives and Tools. Our approach prioritizes simplicity and adaptability, so our application can adapt to different content management systems and domains without costly fine-tuning.

Future work will involve the implementation of more sophisticated AI engineering methodologies to enhance the efficacy of the filter-based paradigm. The goal is to generate more relevant matches and decrease the number of irrelevant ones. For instance, we could incorporate keyword embeddings, similar to those employed in our evaluation, into the application's intelligence to enhance its capacity to more effectively handle synonyms. Furthermore, the system's effectiveness could be enhanced by replacing the current full-text search based on MySQL with dedicated search solutions, such as Apache Solr.

References

- 1 Gaelen P Adam, Jay DeYoung, Alice Paul, Ian J Saldanha, Ethan M Balk, Thomas A Trikalinos, and Byron C Wallace. Literature search sandbox: a large language model that generates search queries for systematic reviews. *JAMIA open*, 7(3):ooae098, 2024. doi:10.1093/jamiaopen/ooae098.
- 2 A. Alharbi, A. Pandit, V. Wilk, P.J. Rosenberger III, and S. Miah. Artificial intelligence-enabled conversational agents in tourism & hospitality: a systematic literature review & future research directions. *Asia Pacific Journal of Tourism Research*, 31(1):21–52, 2026. doi:10.1080/10941665.2025.2555204.
- 3 R. Alotaibi, A. Ali, H. Alharthi, and R. Almehamdi. AI chatbot for tourist recommendations: a case study in the city of Jeddah, Saudi Arabia. *International Journal of Interactive Mobile Technologies*, 14(19):18–30, 2020. doi:10.3991/ijim.v14i19.17201.
- 4 L. Benaddi, C. Ouaddi, A. Jakimi, and B. Ouchao. A systematic review of chatbots: Classification, development, and their impact on tourism. *IEEE Access*, 12:78799–78810, 2024. doi:10.1109/ACCESS.2024.3408108.
- 5 L. Benaddi, C. Ouaddi, A. Souha, A. Jakimi, and B. Ouchao. Chatbots in tourism sector: Classification, evolution, and functionalities. In *Proceedings of the 12th International Symposium on Signal, Image, Video and Communications (ISIVC)*, pages 1–6. IEEE, 2024. doi:10.1109/ISIVC61350.2024.10577889.
- 6 Dimitrios Buhalis. Marketing the competitive destination of the future. *Tourism management*, 21(1):97–116, 2000. doi:10.1016/S0261-5177(99)00095-3.
- 7 Mario Casillo, Federica Clarizia, Giuseppe D’Aniello, Massimo De Santo, Marco Lombardi, and Domenico Santaniello. CHAT-Bot: A cultural heritage aware teller-bot for supporting touristic experiences. *Pattern Recognition Letters*, 131:234–243, 2020. doi:10.1016/j.patrec.2020.01.003.
- 8 Mario Casillo, Massimo De Santo, R. Mosca, et al. An ontology-based chatbot to enhance experiential learning in a cultural heritage scenario. *Frontiers in Artificial Intelligence*, 5, 2022. doi:10.3389/frai.2022.808281.
- 9 Diogo Cosme, Fernando Batista, António Galvão, and Fernando Brito e Abreu. Automating the Classification of Smart Tourism Tools with LLMs: A Taxonomy-Driven Approach. In *Proceedings of the 15th Symposium on Languages, Applications and Technologies (SLATE)*, Open Access Series in Informatics (OASICS), Lisbon, Portugal, June 2026. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. To appear. doi:10.4230/OASICS.SLATE.2026.18.
- 10 European Commission. Leading Examples of Smart Tourism Practices in Europe from the 2025 European Capital of Smart Tourism Competition, 2025. URL: https://smart-tourism-capital.ec.europa.eu/best-practices/european-capital-smart-tourism-best-practices_en.

- 11 António Galvão, Fernando Brito e Abreu, and João Joanaz de Melo. Toward a Consensual Definition for Smart Tourism and Smart Tourism Tools. In *Smart Life and Smart Life Engineering: Current State and Future Vision*, pages 153–183. Springer, 2025. doi:10.1007/978-3-031-75887-4_8.
- 12 Joanna Gotlib-Małkowska, Ilona Cieślak, Mariusz Jaworski, and Mariusz Panczyk. Evidence-based advanced prompt engineering in nursing research: quality analysis of ChatGPT-generated Boolean search query. *Nursing in the 21st Century*, 24(1 (90)):9–19, 2025. doi:10.12923/pielxxiw-2025-0002.
- 13 A. Kontogianni and E. Alepis. AI in smart tourism: LLMs & GPTs leading the way. In *Proceedings of the 5th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–8. IEEE, 2024. doi:10.1109/IISA62523.2024.10786696.
- 14 M.I. Saleh. Generative artificial intelligence in hospitality and tourism: future capabilities, AI prompts and real-world applications. *Journal of Hospitality Marketing & Management*, 34(4):467–498, 2025. doi:10.1080/19368623.2025.2458603.
- 15 SEGITTUR. Catalogue of Technological Solutions for Smart Tourist Destinations – Third Edition. Smart Tourist Destination, 2024. URL: https://www.destinosinteligentes.es/wp-content/uploads/2024/02/Segitur_guia_fichas_2024@14703546813.pdf.
- 16 Giancarlo Sperli. A cultural heritage framework using a Deep Learning based Chatbot for supporting tourist journey. *Expert Systems with Applications*, 183:115277, 2021. doi:10.1016/j.eswa.2021.115277.
- 17 Shuai Wang, Harris Scells, Bevan Koopman, and Guido Zuccon. Can ChatGPT Write a Good Boolean Query for Systematic Review Literature Search? In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR’23)*, pages 1426–1436. ACM, 2023. doi:10.1145/3539618.3591703.

Automating the Classification of Smart Tourism Tools with LLMs

A Taxonomy-Driven Approach

Diogo Cosme¹ ✉ 

ISTAR, Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

Fernando Batista ✉ 

INESC-ID, Lisbon, Portugal

Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

António Galvão ✉ 

CENSE, School of Science and Technology, NOVA University Lisbon, Caparica, Portugal

Fernando Brito e Abreu ✉ 

ISTAR, Instituto Universitário de Lisboa (ISCTE-IUL), Portugal

Abstract

Smart Tourism Tools (STTs) are digital solutions that support tourism stakeholders in designing, managing, promoting, and enhancing tourism experiences. As the number and diversity of these tools increase, consistent classification becomes essential for organizing them in observatory-like repositories and making them easier to search, compare, and reuse. However, previous attempts to classify STTs using Large Language Models (LLMs) showed that broad taxonomy descriptions can lead to subjective and inconsistent interpretations. This paper proposes an improved STT classification framework that operationalizes the original taxonomy as a set of binary classification questions. Each taxonomy category is represented by a yes/no question, making the classification process more explicit, traceable, and easier to analyze. The taxonomy is also refined by renaming selected action domains and categories and removing one low-utility category. An exploratory evaluation was conducted using 100 randomly selected English-language STT descriptions and three state-of-the-art LLMs. A randomly selected subset of 20 examples was validated by a human expert. The results suggest that the proposed framework can support LLM-based classification with substantial alignment to expert judgment, while also revealing remaining challenges related to over-classification and ambiguous category boundaries. The paper contributes a more transparent and reproducible framework for classifying STTs and provides initial evidence of its feasibility for semi-automatic classification in smart tourism observatories.

2012 ACM Subject Classification Computing methodologies → Natural language processing

Keywords and phrases Language Models (LMs), Multi-Label Learning, Domain-Specific NLP, Taxonomy Alignment, Smart Tourism Systems

Digital Object Identifier 10.4230/OASICS.SLATE.2026.18

Category Short Paper

Supplementary Material *Text (Prompts)*: <https://doi.org/10.5281/zenodo.19741250>

Funding This work was partly supported by project 2024.07579.IACDC/2024, funded by FCT – Fundação para a Ciência e a Tecnologia, I.P., under PRR – Plano de Recuperação e Resiliência funding, investment RE-C05-i08 – Ciência Mais Digital and, partially supported by national funds through FCT, ISTAR-Iscte Pluriannual Project UID/04466/2025, and INESC-ID projects UID/50021/2025 and UID/PRR/50021/2025.

¹ Corresponding author



© Diogo Cosme, Fernando Batista, António Galvão, and Fernando Brito e Abreu; licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 18; pp. 18:1–18:11

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Smart tourism has become an increasingly important area of research and practice, driven by the growing use of digital technologies to support destinations, tourism providers, and tourists. Digital solutions can improve the design, management, promotion, and consumption of tourism experiences, while also contributing to more sustainable and data-informed tourism development. However, as the number and diversity of such solutions increase, it becomes more difficult to organize, compare, and reuse information about them in a consistent way.

The concept of Smart Tourism Tools (STTs) was introduced to describe digital solutions that support tourism stakeholders and contribute to smart tourism development. For example, STTs can include booking systems for tourism operators or enable digital experiences within a tourism activity. To support the identification and organization of these tools, a taxonomy of STTs was proposed to classify solutions according to their role in the tourism ecosystem, including their contribution to tourism experiences, marketing, and the design, management, and operation of tourism products [9].

This classification task is particularly relevant to the EUROSTIT observatory², which aims to collect and organize information about STTs from different sources [9]. Since many of these solutions are described in heterogeneous formats, such as catalogues, reports, and promotional texts, manual classification can be time-consuming and difficult to scale. Large Language Models (LLMs) offer a promising way to support this process, since they can interpret natural-language descriptions and assign taxonomy categories with limited task-specific training.

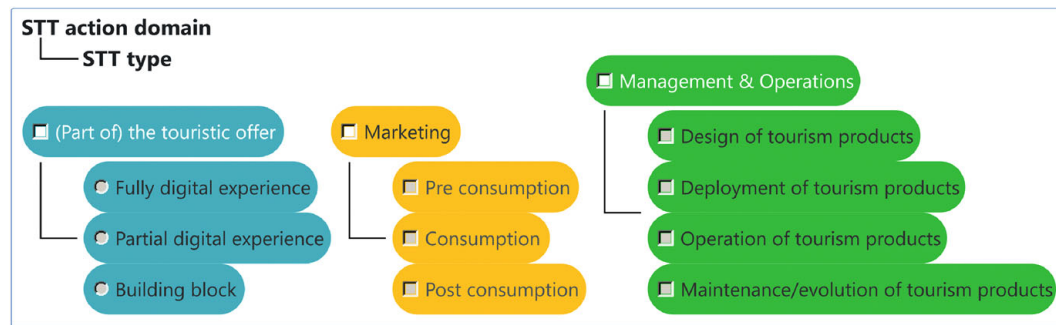
However, previous attempts to classify STTs with LLMs revealed important limitations [6]. The original taxonomy was useful as a conceptual framework, but its category descriptions left room for interpretation. As a result, both human and machine annotators could apply the taxonomy differently, especially when solution descriptions were short, ambiguous, or related to more than one category. This reduced the consistency and explainability of the classification process.

To address this limitation, this paper proposes an improved STT classification framework. The original taxonomy is refined by renaming some action domains and categories, removing one low-utility category, and operationalizing each remaining category through a binary classification question. Instead of asking an annotator or LLM to directly infer the most appropriate category from a broad description, the proposed framework asks whether each category-specific condition is satisfied. This questionnaire-based approach aims to reduce ambiguity, make classification decisions more traceable, and support systematic error analysis.

The paper makes three main contributions. First, it refines the original STT taxonomy to improve clarity and usability. Second, it proposes a questionnaire-based classification framework that transforms taxonomy categories into binary decision questions. Third, it presents an exploratory evaluation of the framework using three state-of-the-art LLMs and expert validation on a randomly selected subset of classified examples.

The remainder of the paper is organized as follows. Section 2 reviews related work on smart tourism, STT classification, taxonomy-based classification, and LLM-based text classification. Section 3 presents the improved classification framework and the revised taxonomy. Section 4 describes the exploratory evaluation with LLMs. Section 5 discusses the main findings and limitations. Finally, Section 6 concludes the paper and outlines directions for future work.

² European Observatory of Smart Tourism Initiatives and Tools



■ **Figure 1** Original taxonomy of Smart Tourism Tools (STTs) proposed in [9].

2 Related Work

Smart tourism has emerged from the increasing integration of information and communication technologies, data-driven services, and interconnected stakeholder ecosystems in tourism. Foundational work on smart tourism emphasizes the role of digital infrastructures, big data, smart technologies, and value co-creation in transforming tourism experiences and destination management [10]. Related work on smart tourism destinations also highlights the importance of technological platforms, real-time information exchange, and personalization for improving tourism services and visitor experiences [2]. More recently, the concept of Smart Tourism Tools (STTs) has been proposed to capture digital solutions that support tourism stakeholders while contributing to sustainability-oriented smart tourism development [9].

The STTs taxonomy [9] was created to address the lack of comprehensive and structured approaches for categorizing smart tourism solutions. The initial version of the taxonomy was developed from the opinions of more than 300 tourism experts collected through a survey. It organized STTs around three main action domains: (i) *(Part of) the touristic offer*, (ii) *Marketing*, and (iii) *Management & Operations*. Figure 1 illustrates the original taxonomy and its respective action domains.

In addition to its conceptual contribution, the STTs taxonomy was designed to support the EUROSTIT observatory, which aims to function as a centralized repository of smart tourism solutions and to improve the visibility of these solutions among Destination Management Organizations, tourism providers, and other tourism stakeholders [9]. Classifying STTs according to this taxonomy provides concrete benefits to these stakeholders. For instance, it empowers Small and Medium-sized Enterprises (SMEs) to bridge the digital divide by allowing them to filter out irrelevant software; a small hotel can easily locate a *Consumption tool*, such as an in-room upselling app, without sifting through back-office logistics software. Similarly, it aids Destination Management Organizations (DMOs) in making strategic, low-risk investments; a DMO tackling overcrowding can use the *Operation* category to quickly discover GDPR-compliant crowd-monitoring sensors for heritage sites. Finally, for the observatory itself, a LLM-based classification framework enables automated market intelligence, allowing the platform to continuously ingest and categorize new digital solutions as they emerge, ensuring the repository remains an up-to-date compass for European smart tourism.

EUROSTIT content is extracted from several sources, including PDF catalogs created by tourism-related organizations [8, 16]. However, such catalogs are often difficult to search, compare, and reuse directly, especially when their content is not structured or easily indexable. As the number of collected solutions grows, the value of the observatory increasingly

depends on the ability to organize STTs consistently according to a meaningful taxonomy. If organized, search is facilitated, namely when using the chatbot our team is developing for this observatory [1].

From a Machine Learning perspective, classifying STTs is a multi-label text classification problem because a single solution may belong to more than one taxonomy category. Multi-label classification has been widely studied, and previous surveys emphasize the specific challenges of modeling label dependencies, evaluating partially correct predictions, and handling imbalanced or ambiguous labels [20]. The STTs classification problem also has a hierarchical component, since labels are organized under broader action domains. Hierarchical classification is known to introduce additional challenges because prediction errors may occur both at the level of broad parent categories and at the level of more specific child categories [17]. These characteristics make the STT classification more complex than the standard single-label document classification.

Traditional text-classification approaches often rely on supervised learning over annotated datasets, including deep-learning methods that learn representations from text and associate them with predefined categories [15]. However, in emerging or highly specialized domains such as smart tourism, large labeled datasets are rarely available. This makes purely supervised approaches difficult to apply in the early stages of taxonomy development. The use of LLMs offers an alternative because these models can perform classification through prompts without requiring a large domain-specific training set [7, 13, 14]. Recent work has shown that LLMs can be effective for text classification tasks, especially when the prompts are carefully designed and when the task benefits from language understanding and reasoning [4, 12, 19]. At the same time, these studies also show that LLM-based classification remains sensitive to task formulation, prompt design, model choice, and the availability of examples.

The use of LLMs in tourism is also gaining attention. Recent studies discuss the role of generative AI, GPT-like systems, and LLMs in tourism and hospitality, including applications related to personalization, service support, content generation, and decision-making [11]. Other tourism-related AI systems, such as chatbots and cultural-heritage assistants, demonstrate the potential of conversational and language-based technologies for improving tourist experiences [3, 18]. Nevertheless, the specific problem of using LLMs to classify smart tourism solutions according to a domain taxonomy remains underexplored.

The first attempt to automate STT classification using LLMs was presented in [6]. In that approach, the taxonomy description and a small number of examples were provided as context, and the LLM was asked to infer the most appropriate classification from each solution description. Although this demonstrated the potential of LLMs for supporting the observatory, several limitations were identified. The task remained highly subjective, the interpretation of category boundaries was not always consistent, and disagreements occurred even among human annotators. As a result, it was difficult to obtain stable classification quality and acceptable inter-annotator agreement.

These limitations suggest that the main challenge is not only the choice of model, but also the way the taxonomy is operationalized for classification. A taxonomy description alone may leave too much room for interpretation, especially when categories overlap or when solution descriptions are short and marketing-oriented. Therefore, the present work complements the STTs taxonomy with a questionnaire-based classification framework. Instead of asking the classifier to directly select categories, each category is represented by a binary question. This design aims to reduce ambiguity, improve traceability, and make classification disagreements easier to analyze. The next section describes the taxonomy refinement process and the resulting questionnaire-based framework.

3 Revised STT Taxonomy and Classification Protocol

3.1 Conceptual Taxonomy

The revised STT taxonomy is a conceptual framework for organizing Smart Tourism Tools according to their primary functional role. The taxonomy is composed of three action domains, each containing a set of categories representing distinct classes of functionality related to tourism. In addition to introducing classification questions, the taxonomy was refined to improve clarity and usability. Some action domains and categories were renamed, and one category was removed.

- The action domains previously called *(Part of) the touristic offer* and *Management & Operations* were renamed, respectively, *Tourism Fruition* and *Design, Management & Operations of Tourism Products* to better reflect the nature of the solutions that fall under these domains.
- The *Building Block* category was removed from the taxonomy because it corresponded to a specific type of solution that was uncommon and, therefore, not useful for classification.
- The categories *Design of tourism products*, *Deployment of tourism products*, *Operation of tourism products*, and *Maintenance/Evolution of tourism products*, were renamed to *Design*, *Deployment*, *Operation*, and *Maintenance/Evolution*, respectively, to make them more concise and easier to understand.

The resulting taxonomy contains three action domains: *Design, Management & Operations of Tourism Products*, *Marketing*, and *Tourism Fruition*. This taxonomy exists independently of the annotation procedure and defines the conceptual constructs used to describe the space of STTs.

3.2 Operationalization as a Classification Protocol

The main objective of the revised framework is to reduce subjectivity and improve the consistency of STT classification. To support more consistent and reproducible annotation, the original taxonomy was retained as the conceptual classification structure, while a questionnaire-based protocol was introduced to operationalize category assignment.

In this protocol, for each taxonomy category, a corresponding binary classification question was defined to guide annotation decisions. When the answer to a decision question is “yes”, the corresponding conceptual category is assigned to the solution. The questions were derived from the descriptions of the original taxonomy categories and designed to make the operational criteria as explicit as possible. This approach transforms the classification task from an open-ended interpretation of category meanings into a series of focused operational decisions. As a result, the framework is intended to support more transparent, reproducible, and explainable classification.

The use of binary questions aims to minimize ambiguity and reduce the range of possible interpretations during classification. In the previous framework, it was often difficult to determine why a given solution had been assigned to a specific category, since the annotator, whether human or machine, could rely on implicit assumptions or different interpretations of the conceptual taxonomy. The questionnaire-based protocol mitigates this issue by making operational criteria explicit. Each category is linked to a specific question, which makes it easier to understand the rationale behind a classification decision. This also facilitates error analysis: when a category is assigned incorrectly, the corresponding question can be examined to determine whether the issue lies in the wording of the question, the interpretation of

the solution description, or the boundaries of the conceptual category itself. Therefore, the protocol not only supports classification but also provides a mechanism for refining the taxonomy over time.

3.3 Classification Questions

The resulting taxonomy contains three action domains and their corresponding categories. To support a consistent application of the taxonomy, each category is paired with a binary classification question used during annotation. The questions shown in Table 1 do not define the taxonomy itself; rather, they operationalize the taxonomy categories into observable decision criteria that can be consistently applied by human annotators or LLMs. For instance, a mobile app that provides AR-guided museum visits and location-aware recommendations would likely satisfy the *Partial Digital Experience* question because it enhances a real-world tourist experience through digital augmentation.

■ **Table 1** Classification Questions for Smart Tourism Tools Taxonomy.

Action Domain	Categories & Questions	
	Category	Question
Design, Management & Operations of Tourism Products	Design	Does the STT help Tourism Providers design and build their offerings?
	Deployment	Does the STT help Tourism Providers deploy their offerings?
	Operation	Does the STT help Tourism Providers manage and operate their offerings?
	Maintenance/ Evolution	Does the STT help Tourism Providers collect and/or analyze data to improve their tourism offerings?
Marketing	Pre Consumption	Does the STT help Tourism Providers promote their services before the tourism experience?
	Consumption	Does the STT help Tourism Providers promote additional offers or services during the tourism experience?
	Post Consumption	Does the STT allow tourists to share their experiences, provide feedback, or act as influencers?
Tourism Fruition	Fully Digital Experience	Is the STT a fully immersive experience within a virtual environment?
	Partial Digital Experience	Is the STT a technological enabler and/or enhancer of the tourist's experience in the real world?

4 Exploratory Evaluation of LLM-based Classification

To assess whether the proposed questionnaire-based framework can be applied automatically by LLMs, we conducted an exploratory evaluation with three state-of-the-art models: *anthropic/claude-sonnet-4.6*, *openai/gpt-5.4*, and *google/gemini-3.1-pro-preview*. The goal of this evaluation was not to establish a definitive benchmark, since no large-scale human-annotated ground truth dataset is currently available, but rather to examine whether the models' classifications were sufficiently aligned with expert judgment to support further development of the approach.

The evaluation dataset consisted of 100 Smart Tourism solution descriptions randomly selected from the available collection. All descriptions used in this experiment were written in English, so no translation step was required. Each model was asked to classify every solution according to the questionnaire-based taxonomy presented in Section 3. Since the taxonomy is multi-label, each solution could be assigned to more than one category. The output format also allowed the models to return no category when the description did not correspond to an STT or when no taxonomy category was applied. The system prompt and classification prompt templates used in the evaluation are available in [5].

From the 100 classified descriptions, a subset of 20 examples was randomly selected for expert validation. For each of these 20 examples, the outputs of the three models were independently assessed by a human expert, resulting in 60 evaluated model outputs. The expert assigned a suitability score using a five-point Likert-type scale: 0, 0.25, 0.5, 0.75, and 1.0. A score of 1.0 indicates that the classification was considered fully suitable, while a score of 0 indicates that the classification was incorrect or unsuitable. Intermediate values were used when the output was partially correct, for example, when the model assigned some relevant categories but also omitted an important one or included a questionable category.

This scoring approach was adopted because the task is not always reducible to a single objectively correct answer. Some solution descriptions are short, ambiguous, or written in broad commercial language, which can make more than one classification plausible. For example, a communication tool that manages customer requests may be interpreted as supporting operations if it is used during service delivery, but it may also have a marketing function if it is used to engage potential visitors before consumption. In such cases, a strict binary correct/incorrect metric would not capture the reasonableness of the model's interpretation. The expert score, therefore, evaluates the suitability of the overall classification rather than only exact label matching.

Figure 2 presents the distribution of the expert scores across the 20 evaluated examples. The best average score was obtained by *google/gemini-3.1-pro-preview* (0.8625), followed by *anthropic/claude-sonnet-4.6* (0.8250) and *openai/gpt-5.4* (0.7250). The median score was 1.0 for all three models, indicating that more than half of the evaluated outputs for each model were considered fully suitable by the expert. However, the standard deviations and the number of zero-score cases show that the models still produced important errors in some cases. Overall, the results suggest that the models were often able to apply the questionnaire-based framework in a way that aligned with expert judgment. Across the 60 evaluated outputs, the overall mean score was 0.8042, and the overall median was 1.0. Moreover, 48 out of the 60 evaluated outputs received a score of at least 0.75, suggesting that most classifications were considered suitable or mostly suitable.

The evaluation also highlights an important boundary condition: the models must not only assign appropriate taxonomy categories but also recognize cases where no category should be assigned. Three examples in the validation subset were judged as not requiring any taxonomy category and were correctly left unclassified by all three models. This suggests that the models can, in some cases, avoid forcing a classification when the description does not correspond to an STT or does not fit the taxonomy. However, this behavior was not consistent. In the case of *Orbis Booking* (id=11), all three models assigned one or more categories, but the expert rated all three outputs as unsuitable. This indicates that false-positive classifications remain a relevant source of error.

The most common issues observed in the validation subset were over-classification and boundary ambiguity. Over-classification occurred when a model assigned additional categories that were only weakly supported by the description. Boundary ambiguity occurred when a

id	name	anthropic claude-sonnet-4.6	openai gpt-5.4	google gemini-3.1-pro-preview
1	ITSVIZIT	0,75	1	0,75
2	Tracking and Digital Antigen Test	0,75	1	1
3	Qred Nomenclator Smart Posters	0,75	0,5	0,75
4	Accessible Pdf Adaptation	1	0	1
5	Transvia Business Chat	0,5	1	1
6	Qfor TOURISM QUALITY	1	1	1
7	Fintech Lab	1	1	1
8	Drafting and Implementation of Covid-19 Contingency Plans	1	1	1
9	2iXR PLATFORM	0,5	0,75	1
10	PLAYVISIT	1	1	1
11	Orbis Booking	0	0	0
12	Seeketing - Mobility and User Behavior	0,25	0,5	0,75
13	Tourist Traceability Sensor	1	1	1
14	Predictive Maintenance of Tourist Assets	1	1	1
15	FS OSINT	1	0	0
16	BEAMBASSADOR	1	0,25	1
17	PHYGI Sensor	1	1	1
18	Smart Occupancy	1	0,75	1
19	APP&TOWN COMPAGNON	1	1	1
20	Chatbot Taro	1	0,75	1
Avg Score		0,83	0,73	0,86

■ **Figure 2** Expert evaluation scores for the LLM-based classification of 20 randomly selected solution descriptions.

solution could plausibly be interpreted as supporting more than one action domain, especially between operational support, marketing, and partial digital experience. These cases confirm the usefulness of the questionnaire-based framework, since each disagreement can be traced back to specific category questions and can therefore inform future refinements of the prompts or of the category definitions.

Given the small size of the expert-validated subset, these results should be interpreted as preliminary. They provide initial evidence that LLMs can apply the proposed framework with substantial alignment to expert judgment, but they do not yet establish generalizable model performance. A larger validation study, ideally involving multiple human annotators and label-level agreement analysis, is required to obtain more robust estimates of classification quality.

5 Discussion

The results suggest that the proposed questionnaire-based framework can make the classification of Smart Tourism Tools more transparent and consistent. By replacing broad category descriptions with binary questions, the framework reduces the space for interpretation and makes each classification decision easier to trace. This is particularly relevant for LLM-based classification, since the model output can be analyzed in relation to specific questions rather than only to final category labels.

Our exploratory evaluation indicates that the tested LLMs were often able to apply the framework in ways that aligned with expert judgment. Most evaluated outputs received high suitability scores, and all three models correctly avoided assigning categories in some cases where no classification was appropriate. This suggests that LLMs can support the semi-automatic organization of STTs in observatory-like repositories.

However, the results also show that classification remains challenging. Some errors were caused by over-classification, where models assigned categories that were only weakly supported by the solution description. Other errors were related to ambiguous boundaries between categories, especially when a tool could plausibly support marketing, operational management, or the tourist experience. These cases show that the framework improves explainability, but does not eliminate the need for expert validation and further refinement. The evaluation should therefore be interpreted as preliminary. The expert-validated sample was limited to 20 randomly selected examples, and only one expert was involved in the validation process. As a result, the findings provide initial evidence of feasibility, but they do not yet establish generalizable performance for the framework or for any specific LLM.

5.1 Threats to Validity and Future Work

Although preliminary results suggest that the questionnaire-based framework improves the transparency of the STT classification, several threats to validity must be acknowledged. First, the exploratory evaluation relied on a limited sample size of 20 solution descriptions. This restricts the statistical generalizability of the findings. Second, a single human expert performed the ground truth validation using a subjective suitability score. Although this metric was useful for assessing partial correctness in an exploratory setting, it introduces potential single-annotator bias and precludes the use of standard machine learning metrics (e.g., F1-score or Hamming Loss).

Furthermore, applying independent binary questions to a hierarchical multi-label taxonomy introduces structural risks, such as potential logical inconsistencies between parent domains and child categories, as well as scalability concerns regarding API latency and token consumption when deployed in large-scale observatory repositories.

To mitigate these limitations, future work will involve a comprehensive empirical validation study. We are planning a synchronous offline classification session with a panel of 30 independent tourism domain experts. To ensure statistical robustness, this study will utilize a targeted overlap design: a dataset of 60 STTs will be structured so that each solution is independently evaluated by exactly five experts. This methodology will yield 300 independent human classifications, allowing for the calculation of Inter-Annotator Agreement (IAA) metrics (e.g., Fleiss' Kappa) to quantitatively measure the framework's effectiveness in reducing human ambiguity.

Establishing this multi-annotator “gold standard” will also transition the evaluation from subjective suitability scores to standard multi-label classification metrics. Finally, to address scalability and structural consistency, future iterations of the framework will explore hierarchical gating mechanisms (e.g., routing sub-category queries only if a parent domain is classified positively) and structured batch prompting.

6 Conclusion and Future Work

This paper presented an improved classification framework for Smart Tourism Tools. The original taxonomy was refined by renaming action domains and categories, removing one low-utility category, and operationalizing the remaining categories through binary classification questions. This questionnaire-based approach aims to reduce ambiguity, improve traceability, and support more consistent classification by humans and LLMs.

An exploratory evaluation was conducted using 100 randomly selected English-language solution descriptions and three state-of-the-art LLMs. A randomly selected subset of 20 examples was evaluated by a human expert. The results suggest that LLMs can often apply the proposed framework with substantial alignment to expert judgment, although errors related to over-classification and category-boundary ambiguity remain.

As mentioned before, future work should expand the validation process to a larger sample and include multiple human annotators to measure inter-annotator agreement and obtain more robust estimates of classification quality. Further work should also evaluate the framework at the level of individual binary questions, refine questions that produce systematic disagreement, and investigate whether examples, prompt variants, or hybrid human-in-the-loop workflows can further improve classification reliability.

References

- 1 Rodrigo Bravo Simões, Adriano Lopes, and Fernando Brito e Abreu. From Natural Language to Structured Queries: An LLM-Powered Chatbot for a Smart Tourism Observatory. In *Proc. of the 15th Symposium on Languages, Applications and Technologies (SLATE)*, Open Access Series in Informatics (OASIs), Lisbon, Portugal, June 2026. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. To appear. doi:10.4230/OASIs.SLATE.2026.17.
- 2 Dimitrios Buhalis and Aditya Amaranggana. Smart Tourism Destinations Enhancing Tourism Experience Through Personalisation of Services. In Iis Tussyadiah and Alessandro Inversini, editors, *Information and Communication Technologies in Tourism 2015*, pages 377–389. Springer, 2015. doi:10.1007/978-3-319-14343-9_28.
- 3 Mario Casillo, Massimo De Santo, R. Mosca, et al. An ontology-based chatbot to enhance experiential learning in a cultural heritage scenario. *Frontiers in Artificial Intelligence*, 5, 2022. doi:10.3389/frai.2022.808281.
- 4 Youngjin Chae and Thomas Davidson. Large Language Models for Text Classification: From Zero-Shot Learning to Instruction-Tuning. *Sociological Methods & Research*, 55(2):501–567, 2026. doi:10.1177/00491241251325243.
- 5 Diogo Cosme. Prompts used in "Automating the Classification of Smart Tourism Tools with LLMs: A Taxonomy-Driven Approach" Paper, 2026. doi:10.5281/zenodo.19741250.
- 6 Diogo F. M. Cosme. Smart ETL and LLM-based contents classification: The european smart tourism tools observatory experience. Master's thesis, Iscte - Instituto Universitário de Lisboa, 2024. URL: <http://hdl.handle.net/10071/34095>.
- 7 Qian Dong, Lei Li, Damai Dai, Ce Zheng, Ke Ma, Rui Li, Yifan Xia, Jingjing Xu, Baobao Wu, Baobao Chang, et al. A Survey on In-Context Learning. *arXiv preprint*, 2024. doi:10.48550/arXiv.2301.00234.
- 8 European Commission. Leading Examples of Smart Tourism Practices in Europe from the 2025 European Capital of Smart Tourism Competition, 2025. URL: https://smart-tourism-capital.ec.europa.eu/best-practices/european-capital-smart-tourism-best-practices_en.
- 9 António Galvão, Fernando Brito e Abreu, and João Joanaz de Melo. Toward a Consensual Definition for Smart Tourism and Smart Tourism Tools. In E. Kornyshova, R. Deneckère, and S. Brinkkemper, editors, *Smart Life and Smart Life Engineering: Current State and Future Vision*, pages 153–183. Springer Nature Switzerland, Cham, 2025. doi:10.1007/978-3-031-75887-4_8.
- 10 Ulrike Gretzel, Marianna Sigala, Zheng Xiang, and Chulmo Koo. Smart tourism: foundations and developments. *Electronic Markets*, 25(3):179–188, 2015. doi:10.1007/s12525-015-0196-8.
- 11 A. Kontogianni and E. Alepis. AI in Smart Tourism: LLMs & GPTs Leading the Way. In *Proc. of the 15th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE, 2024. doi:10.1109/IISA62523.2024.10786696.
- 12 Arina Kostina, Marios D. Dikaiakos, Dimosthenis Stefanidis, and George Pallis. Large Language Models For Text Classification: Case Study And Comprehensive Review. *arXiv preprint*, 2025. doi:10.48550/arXiv.2501.08457.
- 13 Zongqian Li, Yixuan Su, and Nigel Collier. A Survey on Prompt Tuning. *arXiv preprint*, 2025. doi:10.48550/arXiv.2507.06085.

- 14 Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Computing Surveys*, 55(9):1–35, 2023. doi:10.1145/3560815.
- 15 Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep Learning-based Text Classification: A Comprehensive Review. *ACM Computing Surveys*, 54(3):1–40, 2021. doi:10.1145/3439726.
- 16 SEGITTUR. Catalogue of Technological Solutions for Smart Tourist Destinations—Third Edition. Smart Tourist Destination, 2024. URL: https://www.destinosinteligentes.es/wp-content/uploads/2024/02/Segitur_guia_fichas_2024@14703546813.pdf.
- 17 Carlos N. Silla and Alex A. Freitas. A Survey of Hierarchical Classification Across Different Application Domains. *Data Mining and Knowledge Discovery*, 22(1):31–72, 2011. doi:10.1007/s10618-010-0175-9.
- 18 Giancarlo Sperlí. A cultural heritage framework using a Deep Learning based Chatbot for supporting tourist journey. *Expert Systems with Applications*, 183:115277, 2021. doi:10.1016/j.eswa.2021.115277.
- 19 Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. Text Classification via Large Language Models. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8990–9005, Singapore, 2023. Association for Computational Linguistics. doi:10.18653/v1/2023.findings-emnlp.603.
- 20 Min-Ling Zhang and Zhi-Hua Zhou. A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014. doi:10.1109/TKDE.2013.39.

Lazarus: Where Compilers Meet Education

Rui Gonçalves ✉ 

ALGORITMI Research Centre / LASI, Dept. of Informatics, University of Minho, Braga, Portugal

Pedro Rangel Henriques ✉ 

ALGORITMI Research Centre / LASI, Dept. of Informatics, University of Minho, Braga, Portugal

José Carlos Ramalho ✉ 

ALGORITMI Research Centre / LASI, Dept. of Informatics, University of Minho, Braga, Portugal

Abstract

Lazarus is an intentionally compact platform featuring an optimizing compiler, a lightweight virtual machine and IDE integration. Its small size, lets students explore every compilation stage by experimenting with a compiler inspired by modern systems. At the core lies an incremental parser, that only re-parses the surrounding areas of the changed text in an IDE environment, and an SSA-based optimizer. The target machine is a register-based virtual machine, that mimics native architectures, and provides a debugger. This paper presents both the source language, and its register-based virtual machine, while discussing an often neglected feature in compiler design: the incremental parser.

2012 ACM Subject Classification Software and its engineering → Object oriented languages; Software and its engineering → Parsers; Software and its engineering → Compilers; Software and its engineering → Source code generation; Software and its engineering → Virtual machines

Keywords and phrases Object-Oriented Programming, Compilation, SSA, Optimization, Virtual Machine, IDE

Digital Object Identifier 10.4230/OASICS.SLATE.2026.19

Supplementary Material *Software (Web Site)*: <https://lazarus.ep1.di.uminho.pt>

Funding This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Unit Project Scope UID/00319/2025 – Centro ALGORITMI (ALGORITMI/UM) <https://doi.org/10.54499/UID/00319/2025>.

Acknowledgements The authors would like to thank Neal Gafter and Eric Lippert (from the Roslyn team) for their valuable support, for answering numerous questions, and enabling some features presented in this work; and acknowledge the support and help of our families during this time.

1 Introduction

Computer science degrees often include courses on language design and implementation intended to instruct students on the process of specifying formal languages through the usage of context-free grammars, as well as the processes used to analyze and handle source code. Under this context, programming languages are usually the subject of a course's practical work, although in a simplified form, such as subsets of C or Forth. Given the inherent theory and complexity of compilers, which are not the focus of the class, they are often neglected and replaced by abstract models, such as Intermediate Representations (IRs) and Stack-Based Virtual Machines. While these serve their purpose in education and non-academic environments, the backend components and processes mostly remain as black boxes, creating a very large gap in understanding.

Furthermore, in recent decades, “Programming Languages” have undergone drastic changes facilitated by increased computational resources. Compilers can now (and are expected to) empower code editors with sophisticated tooling tied to the language ecosystem,



© Rui Gonçalves, Pedro Rangel Henriques, and José Carlos Ramalho;
licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 19; pp. 19:1–19:18

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

such as providing type information, inline documentation, real-time diagnostics, auto-fix suggestions, auto-completion, refactoring primitives, and much more. Moreover, this tooling is the cornerstone of a new wave of revolution in the programming language and framework ecosystems. It is directly exposed via MCP (Model Context Protocol) [2, 3] to AI agents, such as Dart (and Flutter) MCP [24], or in other novel approaches where languages are built specifically for this purpose: the so-called AI-native languages like MoonBit [18].

This shift has created a significant architectural divergence compared to the academic models taught in these courses or classical compiler textbooks regarded as the “gold standard” worldwide, such as *The Dragon Book* [1] and Appel [4]; these are now quickly becoming out-of-sync with industry practice. This issue is even more severe in other areas of the spectrum: optimization, machine code generation, debugging, etc., because of their increased complexity and lower-level nature. Moreover, while many compilers rely on the LLVM infrastructure [32, 34], which is not beginner-friendly to study and modify without prior knowledge and experience. This led to the creation of a simpler, smaller, and easier-to-modify programming language infrastructure introduced in this paper.

This paper is organized in seven sections. Section 1 presents the project context and motivation. Section 2 provides an overview of related work and industry practice. Section 3 presents a brief description of the source language. Section 4 goes into detail of the architecture of the virtual machine. Section 5 introduces the compiler/infrastructure architecture. Section 6 explores the incremental parser algorithm. Finally, section 7 closes the paper with the conclusions and possible future directions for the project.

2 State of the Art

A compiler is divided into three main parts: the front-end, middle-end, and back-end. The front-end deals with the source language itself; it *reads* the input text (source program), validates it, and extracts its semantic meaning. The middle-end performs optimization passes on the source program aimed at improving the program execution performance. The back-end is responsible for producing assembly code. To complete the Compilation process, there is still the need for an assembler that translates the assembly code generated by the compiler into an object file (LLVM actually bundles the assembler and produces the object file directly from its representation, skipping the translations). Additionally, a mechanism to resolve external symbols is needed, such as a linker in C or dynamic resolution at runtime as in Java. For compiled programs to be executed, they need a runtime environment, such as an operating system or virtual machine, as well as standard libraries that might have been linked with.

2.1 Compiler Front-End

The front-end is the component used for the interface with code editors. Microsoft developed the Language Server Protocol (LSP) [38] as a standardized way for different code editors to request common and expected features from the front-end. Moreover, since the source code document is constantly changing and those changes must propagate to the front-end, meaning the front-end itself must also efficiently represent text in a format that is both easy and convenient for the analysis it performs as well as for user editing. Text editors usually use variants of *Line Arrays*, *Piece Tables*, *Gap Buffers*, and *Ropes* [36, 7, 6]; which may be augmented by extra metadata to help with line breaks and encoding formats. *Line Arrays* keep an array where each position is a string representation of the line. *Gap Buffers* operate on the premise that multiple consecutive user operations are performed around the

same position, and as such when an insert operation is performed it allocates space for the successive insertions. *Piece Tables* try to optimize a requirement of code editors, the undo operation, and essentially keeps the original source and tracks the changes, requiring multiple jumps to query the up-to-date source, but reducing overall editor memory. *Ropes* on the other hand operate on balanced tree structures, making an insertion a join of nodes, this structure has the great advantage of immutability, making it one of the ideal options for front-ends.

The first step in a front-end is to take the input textual stream and produce tokens (like words in English), this is lexical analysis. Then comes parsing, checks the token sequence against grammar rules and produces a representation of the input program in a tree form: the syntax tree. Traditionally, this tree removes redundant information (e.g., spacing, comments, parentheses around conditions, or in math expressions), creating an abstract (not concrete) representation: the Abstract Syntax Tree (AST). On the other side of the spectrum, Roslyn (the C# compiler) [37] produces a tree containing every syntactical detail (a 1:1 mapping), known as Concrete Syntax Tree (CST). Compilers such as Go tend to opt for a middle ground; the tree still contains details like parentheses, and comments (kept in a separate table in Go), but spacing, line breaks, and other non-essential information for the compiler are discarded.

To create a parser, a language must first be described by a grammar, which defines the productions (rules) used both for generation (the derivation of language sentences) and for recognition. The rules make use of tokens (terminal symbols), and non-terminal symbols. Two main core parsing strategies exist: one is Top-Down (TD) – usually implemented by parsing algorithms of the LL family – where the structure is built from the beginning (top of the tree) to the end (bottom); for example, a sentence must contain a subject and a predicate, and so on down to the word. The other is Bottom-Up (BU) – usually implemented by parsing algorithms of the LR family – that analyze input terminals until recognize the right-side of a production reducing, at this moment, the input sequence to the non-terminal in the left-side of that production; and proceed so far, so on, until be able to reduce the complete input sequence to a non-terminal that is the root of the tree. For example, a name is a subject, and the subject plus a predicate form a sentence (the sentence is only recognized (accepted) when both elements are found). The needed grammars can be represented in common formats such as EBNF [26], which will be adopted throughout this paper.

Given that creating a lexer/parser for a set of grammar rules is a systematic and repetitive work, several amazing parser (and lexer) generators were developed. The most famous ones are Lex/Yacc (the eldest, inspirational, and the most used around the world by the C community¹), Tree Sitter [9] (an incremental parsing library/generator developed for the Atom code editor based on Generalized LR theory) and ANTLR [43] (Another Tool for Language Recognition based on Adaptive LL theory)². Incremental parsers are those capable of being aware of a change (from a code editor) and producing a brand-new CST by piecing together a previously built CST with minimal lexing/parsing, covering just the changed region. While these generators are very capable, most commercial-grade compilers end up handwriting a Recursive Descent Parser (follows the TD strategy, but resorts to several tricks to solve ambiguities, including but not limited to arbitrary look-ahead, LR-like behavior, and semantic/syntactical aware parsing/lexing). The choice for RD parsers happens because they are easy to make, capable of better/custom error recovery and messages, and feature gating – the ability to control access to language features, as is the case of versioning, variants (e.g., a single compiler for all C-family languages), enabling experimental language features.

¹ Nowadays available for Python as a module: PLY.

² A similar version is also, at present, available for the Python community as a module: Lark.

The next step is semantic analysis or binding, where the syntax tree is processed and types and other contextual conditions (semantic constraints) are checked. Most compilers tend to store this information in one of two places: the syntax tree itself (also called the annotated syntax tree) or by producing a brand-new bound tree. Most compilers end up doing multiple passes through the entire tree, performing different semantic analysis tasks: declaring classes, fields, and methods; then expression-level type checking. In these pipelines, each step creates/augments a symbol based on previously known symbols (variables, functions, classes, or anything relevant) and stores them in a centralized structure: the symbol table. Symbols generally use lazy evaluation, and there are multiple passes to avoid ordering problems (e.g., a function using another function declared later). In recent times, several compilers have also adopted Hindley-Milner [25, 40] based type systems that employ constraint solving systems to automatically deduce types. Moreover, most compilers cache the values computed along these passes (or parts of the passes; or much more simply on a per file/library basis) so they can be used for incremental semantic analysis.

2.2 Compiler Middle-End

For the rest of the compilation pipeline, most brand-new compilers tend to rely on the LLVM infrastructure, although MSVC, GCC, Go, and other compilers still prefer to build their own (for better compilation time, flexibility, simplicity, and other reasons). These systems work on an Intermediate Representation (IR) that the front-end must emit, to allow for optimizations to be constructed on a common infrastructure and tooling. In the past, IRs were high-level and quite similar to the source language, but nowadays most IRs opt for being low-level. This generalizes and canonicalizes the source, making it uniform (no semantic-specific constructs), enabling easier reasoning and transformation, which in turn improves the correctness of optimizations and the generalization (more powerful) and composability of them. Furthermore, some optimizations always require target-specific information (e.g., vectorization); as such, it is common to specify target details, such as architecture, extensions, operating system, and others; and pass them along the optimization pipeline.

IRs nowadays also use sparse representation that encode directly variable *def-use chains* (what variable uses a certain definition) and *use-def chains*. This is augmented by variable immutability (and a special Phi function for merging control flow branches), known as Static Single Assignment (SSA) form [16, 45]. To convert to this form, it is necessary to insert Phi instructions at control flow join points to merge incoming values; the set of nodes where such insertions may be required is the dominance frontier [44, 35]. This set can be calculated in multiple ways and may use sophisticated algorithms and data structures such as the DJ-Graph [47, 33]. Given the complexity of this task, LLVM introduced a convenient way for front-ends not to have to implement this logic every time; it uses a “memory-based” form (stack allocations and read/write operations) whose variables are then promoted to SSA registers via the *mem2reg* pass. This form is also convenient for calculating liveness information, which indicates what variables are live at the end of a block (needed by a successor block) and in the beginning of a block (used in the block or a successor), as well as other data-flow analyzes.

Most compilers also implement hundreds of small optimizations for different use cases and patterns, making their study an impossible task. As such, only a handful of optimizations considered to be some of the main ones will be discussed. Sparse Conditional Constant Propagation [51] performs a form of Dead Code Elimination (removing unused branches/variables) and evaluates constant expressions together to improve the accuracy of constant value detection. Global Value Numbering (GVN)/Global Code Motion (GCM) [10] merge redund-

ant operations that compute the same value and hoist code to better places (e.g., outside loops), performing a form of redundancy elimination and loop-invariant code motion. More general forms of redundancy elimination, such as partial redundancy, where a computation may occur in one but not all execution paths, can augment the GVN algorithm [48]. This is the reasoning behind the deprecation of more classical algorithms for the task, such as SSAPRE [41, 27, 28]. With the widespread use of streams in languages, algorithms have also been developed to perform stream fusion, which combines multiple passes over data into a single pass and conversion to standard imperative constructs [29].

The question of the order of optimizations led to the combination of different sorts of optimizations into singular passes, in an attempt to improve compilation speed and effectiveness [13, 15]. Moreover, GVN/GCM optimizations require a particular internal structure: a graph where the notion of definitions is discarded since edges represent usage. This internal structure (which still holds to the SSA property) has been adopted as the core intermediate representation of some compilers and is named the Sea of Nodes [14, 11, 12]. This representation led to the development of new SSA conversion algorithms that require no prior analysis [8], and more advanced peephole optimizations (optimizations requiring no prior analysis), enabling optimizing one-pass front-ends where parsing directly produces the IR (which is automatically optimized on creation), and type checking is also performed on the IR itself before iterating through more advanced optimizations. This approach generally produces much higher-quality code faster, but its main disadvantage for traditional compilers is that GCM is non-optionally integrated, making debugging much more difficult.

2.3 Compiler Target Machine

Before continuing with the pipeline it is important to better understand the general target of a compiler. A CPU or virtual machine can only execute a defined set of instructions, which operate on specific data types and registers and access memory in particular ways [42]. All instructions must also follow a specific encoding format, and the Instruction Set Architecture (ISA) defines all of these aspects. On top of the hardware layer itself, there is usually an operating system [5]; for all intents and purposes, it is just like a big virtual machine that exposes an interface to access system resources via system calls. The program itself is not just code; it also contains other information such as imported/exported symbols, libraries to load, signatures, debug information, and other metadata. That referred information is stored in the image/container format (e.g., PE (.exe/.dll) in Windows and ELF in Linux/other Unix-like systems). Moreover, there are several other details that need to be specified, such as naming conventions, calling conventions, stack layout, exception models, among others. These details about how a compiled program interacts with the OS, hardware, and other binaries are called the Application Binary Interface (ABI).

Virtual machines provide the ability to run isolated environments of an ABI that may correspond to a physical ABI (e.g., x86_64, ARM) or a completely virtual one. Several compilers for well-known programming languages (e.g., Java, C#, F#, Visual Basic, Python, etc.) resort to a virtual machine approach for some of its advantages, such as *portability* – which Java popularized with the “write once run everywhere” slogan – or *performance* reasons (in comparison to tree-walk interpreters). These virtual machines can include extra facilities compared to real architectures, such as an unlimited number of registers or “locals”, or no registers at all (just a stack). Furthermore, the virtual machine can be augmented to compile the running application to the native architecture on demand. Essentially, this “bundles a compiler” (a Just-in-Time (JIT) Compiler) which can use its knowledge about the running program (e.g., frequently executed paths) that would not be known without runtime support, to *safely* optimize the program more aggressively than an ahead-of-time compiler ever could.

Beyond the image that the runtime environment executes, there is an additional layer of metadata designed to enable debugging. This information provides a mapping between logical language constructs (e.g., variables and parameters) and physical ones (e.g., memory/stack locations and registers), as well as mapping addresses to line information, among others [17]. Virtual machines allow for much richer debug information since they do not have the constraints of physical architectures. This flexibility enables specialized ways to simplify this mapping, such as custom debug-specific instructions that provide fine-grained information, or even high-level constructs such as classes being delayed into the runtime with all their information. Furthermore, virtual machines typically execute de-optimized code during debugging, rolling back from highly optimized code that is executing. This allows the usage of an almost direct correspondence from the source code, significantly enhancing the debugging experience and enabling more complex features [31, 30]. While it is technically possible to achieve the same in native architectures, doing so would introduce substantial instrumentation overhead, even more than the already present one (the operating system’s middle-man overhead). To bridge the gap between debuggers and code editors, there are several protocols, such as the Debug Adapter Protocol (DAP) [39], that standardize common functionality; but given the versatility of some debuggers and the features they provide, the usage of these protocols can be limited. Certain debuggers provide powerful tools – such as, time-travel debugging (stepping backwards), predictive debugging (seeing future execution paths when execution is paused), custom data visualizers (displaying 3D models) – which are highly specific to certain use cases of that specific debugger and cannot be expected to be standardized.

2.4 Compiler Back-end

Returning to the compilation pipeline, the back-end must take the mostly architecture-agnostic IR and produce the final target-dependent assembly. The journey begins with instruction selection, where IR instructions are converted into their target equivalents. This task presents some interesting aspects. The first is the possibility of a non-direct mapping: a singular native instruction might merge multiple IR ones (e.g., in x86_64, the Load Effective Address instruction is usually used to replace multiple arithmetic operations that follow the pattern $base + index * scale + offset$), or vice versa. The other interesting detail is that the order in which instructions are presented to the CPU might affect performance (because of hardware details like pipelining). It is very common to perform scheduling after selection for this purpose, or to take these patterns into account during selection itself. Conceptually speaking, this step is quite simple, and even simple tree-pattern matching [19] can do it. Register allocation usually follows, and this is arguably the most intricate step of the back-end. In this step, the compiler must assign a place to variables, and when there is no room for the variables, they must be temporarily stored in the memory stack. This problem is NP-complete, so heuristics and other mechanisms are used. There are some simplistic algorithms that yield excellent results, like Linear Scan [47], which have been used successfully in both compilers and JIT-based virtual machines. Other approaches involve the equivalence to the graph-coloring problem and do exactly that, such as Chaitin-Briggs style, Iterated Register Allocation [22, 46], and some of its generalizations. Other approaches even take advantage of the SSA property and use chordal graphs to solve the problem. This last approach raises an interesting question: where is the best place to destroy (remove) SSA? This destruction process involves replacing `Phi`s with copy instructions on incoming blocks, and some compilers prefer to perform this even before instruction selection. Although destroying during the register allocation process has clear advantages for analysis, and some

back-ends opt for this approach as well. While there are always variations to the pipeline described, and every single compiler ends up making significant architectural changes to it, this is especially true in the back-end.

3 Language

The high-level programming language to be compiled, named Lazarus (after the infrastructure³), is designed as a statically typed, imperative, and object-oriented programming language with a focus on familiarity and simplicity. The built-in atomic types include **String**, **Number**, and **Boolean**. The numbers are represented by double-precision floating-point values (IEEE 754), just like JavaScript. Moreover, the language has array support and is also garbage collected. The standard control-flow structures are supported, such as the **if-else**, **for**, **while** and **do** statements. The variables are mostly type-inferred, although the return type and argument types still require explicit type declarations. The language also provides some extra native functionality for writing to the console and some basic graphical capabilities. It uses braces and semicolons, allowing for an easy transition from C# (the language in which the infrastructure was written), although it deviates from some syntax choices of C-style to simplify the compiler itself. The most significant change is found in the generic syntax, which removes the *less than* (<) and *greater than* (>) symbols, given that they would introduce ambiguities into the language. The generic support is used to provide the built-in types, in particular the `Nullable@(<T>)` and `Array@(<T>)`. Given the large scope of the language, only a small subset is described. Currently, the language accepts only a single input file, which must be composed of declarations (classes or functions), described by the top level grammar in Listing 1.

■ **Listing 1** Top-level language grammar.

```

ClassDeclaration →
| "class" ID GenericParameterDeclaration? ClassExtends? ClassBody
ClassExtends → "extends" TypeExpression
ClassBody → "{" ClassMember* "}"
ClassMember → Field | Method
Field → ID Typing ";"
Method → ("virtual" | "override")? Function
FunctionDeclaration → "fun" Function
Function →
| ID GenericParameterDeclaration? "(" ParameterList? ")" Typing? Block
ParameterList → Parameter ("," Parameter)*
Parameter → ID Typing
Typing → ":" TypeExpression
TypeExpression →
| "fun" "(" TypeExpressionList? ")" Typing? "?"?
| ID GenericArgument? "?"?
TypeExpressionList → TypeExpression ("," TypeExpression)*
GenericParameterDeclaration → "@" "(" ID ("," ID)* ")"
GenericArgument → "@" "(" TypeExpression ("," TypeExpression)* ")"

```

³ It came to our attention that Lazarus is an established name in the industry, yet we opted to retain it, as it was chosen as an allusion to the biblical narrative, and the project's revival from a prior attempt.

19:8 Lazarus: Where Compilers Meet Education

The language intentionally leaves out access modifiers, which are helpful in large-scale systems but mostly useless in small-scale “scripts”, which this language aims for. Having said that, by convention, fields starting with the underscore character are considered private (but are accessible nonetheless). The only modifiers available in the language are *virtual* and *override*, which are applicable only to methods. A virtual method indicates that the method can be overwritten by another class; such a class must specify the intent using the *override* keyword. This choice allows for exact control over the methods present in the tables required for virtual dispatching. The typing system also natively uses nullable types, which is syntactic sugar for the *Nullable@(*T*)* class type, and function pointers, although it can be easily extended to support closures down the road. Regarding generics, generic arguments cannot be refined as of now, since there is no interface or trait support; this presents a challenge for real-use cases of the language.

Listing 2 shows a simple program written in Lazarus to illustrate the basics of the language. The example is a function (the main one) to print a header text and the numbers between 1 and 5, one per line, using a “for” loop. Listing 3 shows a more complex example, by computing the number of primes from 1 to 20 using the Eratosthenes Sieve.

■ **Listing 2** Minimal program sample.

```
fun main() {
    print("Hello World!\n");

    for (var i = 1; i <= 5; i = i + 1) {
        print("Number $i\n");
    }
}
```

■ **Listing 3** Eratosthenes Sieve in Lazarus.

```
class PrimeEratosthenesSieve {
    n: Number;
    isPrime: Array@(Boolean);

    init(nbr: Number) {
        n = nbr;
        isPrime = new Array@(Boolean)(n + 1, true);
        isPrime.set(0, false);
        isPrime.set(1, false);
    }

    run() {
        for (var i = 2; i * i <= n; i = i + 1) {
            if (!isPrime.at(i)) continue;

            for (var j = i * i; j <= n; j = j + i)
                isPrime.set(j, false);
        }
    }

    countPrimes(): Number {
        var count = 0;

        for (var i = 0; i < isPrime.length; i = i + 1)
            if (isPrime.at(i))
```

```
        count = count + 1;

    return count;
}

fun main() {
    var nbr = 20;
    var sieve = new PrimeEratosthenesSieve(nbr);
    sieve.run();
    var count = sieve.countPrimes();

    print("There are $count primes from 1 to $nbr");
}
```

3.1 Language Analysis

The language itself is parsed by a handwritten recursive descent parser, which makes use of a precedence table for correctly parsing expressions. Moreover, this parser employs a custom error recovery mechanism that attempts to never consume tokens beyond a statement, by keeping a global recovery synchronization set with the token that close and open new statements, and not crossing that boundary when already in panic mode. The parser produces a CST (Concrete Syntax Tree) [23], by keeping every error and whitespace (trivia tokens) attached the tokens in the tree, enabling the code formatter and the incremental parser, improving the developer experience and productivity. The exact internals of the incremental nature of the parsing algorithm will be discussed later.

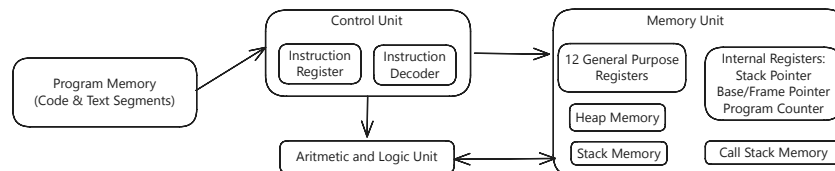
The semantic analysis is composed of multiple passes. First, the class declaration pass declares all available classes, making it possible to reference them later even if their declaration is present after usage. Next is the class inheritance pass, which maps and checks the possibility of the class hierarchy. Symbol information is created in a lazy fashion; meaning generic class fields and methods are only physically created when they are queried, making it possible to write code such as `class A extends B@C` even though the fields and methods were not declared (this occurs in the next pass). Moreover, since virtual functions might not have been declared, it is not possible to determine if an override method is possible; therefore, a following pass dedicated solely to these checks is necessary. Last, the function pass performs type checking and produces a bound tree with this information.

4 Virtual Machine

The Virtual Machine was designed to be a lightweight alternative to native architectures, allowing cross-platform execution without heavy emulators. Moreover, this approach proves useful as it allows for the study of debuggers without relying on operating systems and their complex debugging APIs. This machine is register-based; most instructions cannot operate on memory directly, requiring values to be loaded into registers or being immediate. Furthermore, the program has access to a stack where temporary values (e.g., spilled variables) can be placed and arguments can be passed to functions with a variable number of parameters (which are called variadics). While this stack is traditionally also used for call management in native architectures, it was opted to separate it into an implicit stack, ensuring that a program cannot accidentally corrupt this information and minimizing function prologues and epilogues. The program can also allocate memory on the heap; in which case, the memory is automatically garbage collected. Since no symbolic names are used by the virtual

19:10 Lazarus: Where Compilers Meet Education

machine, access to data in these heap-allocated structs must be done by offsetting the base value. Moreover, since all kinds of values are considered boxed (reference types), this offset corresponds to the field index. A program itself has multiple sections, mainly the code and data sections, which contain the instructions to be executed and constant strings respectively. This data section can be extended in the future to allow for constant data besides strings to enable jump tables commonly used by switch statements, virtual dispatch tables (which are currently created at instantiation time), richer typing tables, among other common use cases. Figure 1 contains a visual representation of the described architecture.



■ **Figure 1** Virtual Machine Architecture Diagram.

The Instruction Set Architecture of this machine contains several opcodes to perform standard **arithmetic and logic operations**, to perform **calling and returning**, **nullability checks**, **stack management**, **jumping**, to **allocate** and **operate on heap structures**, and some extra special instructions meant to interface with the user (the **system call** instruction) and to call the **debugger** (equivalent to the *int3* instruction in x86, which is used to call the debug exception handler and is inserted by the debugger; this will be explained later). The encoding schema for the instructions has a variable length depending on the variant of the instruction. An instruction can contain *up to three arguments*, and only *two of them can vary* (such as *immediate* or *register*). This variant information is stored in the two upper bits, leaving the rest of the byte for the opcode; this, in turn, enables up to $2^6 = 64$ opcodes. The actual argument encoding follows this byte, and immediate values are stored in little-endian order. The architecture follows a Reduced Instruction Set Architecture (RISC) style, preferring simple instructions rather than more compact and efficient ones. In the future, this machine can be extended to support a second register class for vector registers and the corresponding arithmetic and logic instructions to operate on them, enabling the study of Single Instruction Multiple Data (SIMD) used by vectorization optimizations. The compiled code is loaded from a container format that can hold more than the two aforementioned sections (the code and strings). These two extra sections hold linking and debugging information. The linking section contains information about exported symbols and imported symbols, enabling the linker to effectively resolve locations and produce an executable. The debug tables contain line, variable, typing, and other information based on a subset of the DWARF standard. Listing 4 contains an implementation in the virtual machine assembly for the program in Listing 2.

■ **Listing 4** Minimal assembly sample.

```
.strings
    "Hello World!\n"
    "Number "
    "\n"
.end

entry main:
    mov #0 r8          ; moves "Hello World!\n" to register 8
    syscall 0          ; call print (register 8 is printed)
    mov 1 r9           ; i = 1
```

```

loopCond:
    leF64 r9 5 r8      ; i <= 5
    jumpIf @loopBody @loopExit r8
loopBody:
    mov #1 r8          ; moves "Number " to register 8
    f64ToStr r9 r10     ; converts variable i to string
    addStr r8 r10 r8     ; interpolates the string
    mov #2 r10          ; moves "\n" to register 10
    addStr r8 r10 r8     ; interpolates the string
    syscall 0           ; call print
    addF64 r9 1 r9      ; i = i + 1
    jump loopCond
loopExit:
    return

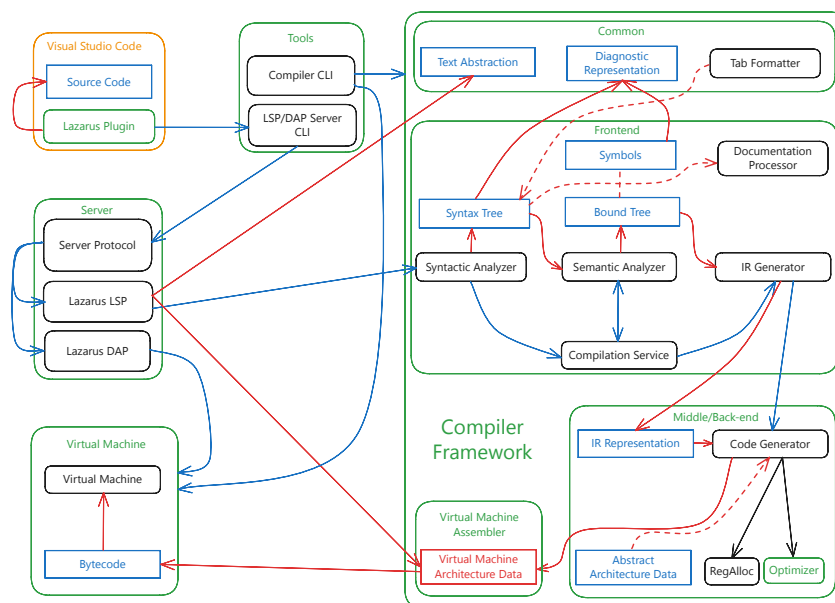
```

The virtual machine’s debugging approach also mimics that of production-ready native debuggers. The typical way these debuggers handle breakpoints is by replacing the instruction with an interrupt (like *int3* in x86, already mentioned). When the Control Unit hits that program instruction, the operating system is called and transfers execution to the debugger. That instruction must be the shortest possible (*int3* is one byte) to ensure it does not overlap with a second instruction, while the original is kept in the debugger, ensuring it can then continue execution. This virtual machine uses the same approach, just streamlined by the fact that there is no middleman between the debuggee and the debugger. Standard stepping is implemented based on the same principles: *step in* executes instructions until the line changes, while *step over* does the same but also decodes the current instruction to take into account calls. Native debuggers sometimes end up disassembling the machine code and speculating where the control flow is going to end (faster than single-step instruction execution because of operating system overhead), which was again streamlined in this debugger. *Step out* is the most interesting one: it is not possible to simply take the return address and insert a temporary breakpoint there because of recursion; so, instead, it modifies the call stack value to a virtual machine-controlled address (a breakpoint instruction outside code space) so that when hit, the debugger knows it can replace the program counter. Native architectures use a similar approach but tend to replace it with an invalid address instead (since the operating system notifies the debugger).

5 Architecture

The compiler front-end follows the same architecture as Roslyn and is quite similar in many details, with several internal APIs being almost identical. The middle-end and back-end were inspired by LLVM, but the chosen architecture significantly differs from it. Figure 2 contains a diagram of the high-level overview of the final architecture of the compiler infrastructure. The “Common” submodule contains the internal data structures for text representation and diagnostic information. These structures are used by the rest of the infrastructure, including the actual LSP/DAP, which automatically translates to/from the protocol specific to each. The Frontend submodule contains everything related to the Lazarus language itself, ranging from representations to the documentation processing engine, and some services, such as the code formatter. The Middle/Back-end contains the IR representation, as well as the implementation of different optimization passes and machine code generation. The virtual machine assembler implements the architecture-specific parts for the backend, such as defining available registers and instructions and producing the actual machine code. Lastly,

the virtual machine submodule contains the execution and debugging engines as well as the bytecode representation. Complementary to the Compiler Framework is the suite of tools containing the Command Line Interface (CLI) (*lz*) to run and execute Lazarus code from the terminal, as well as the LSP/DAP server (*lzs*). The latter tool uses the Server module, which contains a generic implementation of the Base Protocol, the LSP and DAP protocols, enabling code reuse, as well as the Lazarus-specific implementation. Lastly, the Plugin module contains the Visual Studio Code-specific code for launching *lzs* and connecting to it as a client. The LSP uses some custom messages to provide extra functionality, such as the syntax tree visualizer, the control flow graph viewer, and the optimization pipeline inspector. These extra tools provide direct access to internal structures used within the compiler for several tasks, enhancing understanding of the inner workings by streamlining their visualizations, and serving as debug tools for compiler engineers.



■ **Figure 2** Lazarus Infrastructure Architecture Diagram.

The entire architecture was also built around the principle of providing access to the internals of the system in a more visual and easy to understand way, which is accomplished by exposing this information via Visual Studio Code extension. For brevity, it is only going to be enumerated some of them. The syntax tree is exposed to the user, and it highlights the nodes changed by the incremental parser. The entirety of the pipeline stages (i.e., optimizations, and code generation) are exposed in a textual form that highlights the changes. A function's control flow graph used by the bound tree can be displayed.

6 Incremental Parser

As already mentioned, an incremental parser is the technology that enables efficient re-use of previous Syntax Trees, by only reparsing changed regions of the source code; making a crucial feature for the compiler's integration in an IDE. The system is built on top of two very important properties of the **Concrete** Syntax Tree. The first one being immutability. After creation, it is not possible to change the tree, hence the usage of bound trees, making it possible to completely reuse subtrees in other places, as is the case with the refactoring tools,

the code formatter, and of course the incremental parser. Another advantage of this property is trivial concurrency, although it has not been exploited by the compiler yet. The second property is that the used tree is actually not the one produced by the parser. If every node knew its exact position in the source code (i.e., line and column information), any edit action would force rebuilding the entire parse tree after the change. Instead, the parser produces a persistent tree, which is informally called the “green” tree, that keeps track of the width of the node and not its position. Since knowing the exact position is often required, a new tree (the “red” tree) is created, which is just a thin wrapper over the green one, and does the position calculations [49, 50]. This red tree is not fully created at once; doing so would negate the effects of the incremental parser, but rather when a child of a node is accessed, all the children of that node are created. Since not all parsed trees need to be traversed in their entirety, for instance, a new change has been received while the user is typing, not all red tree’s nodes are going to be fully created. While the actual implementation uses an index based position, which simplifies the algorithm, it can be modified to also include line and column spans, enabling the red tree to present the position in a more convenient format; though the current approach just computes this information on-request and is incrementally updated by the text representation (a modified rope data structure with line and column cache). The definitions of these two trees are generated by a Source Generator.

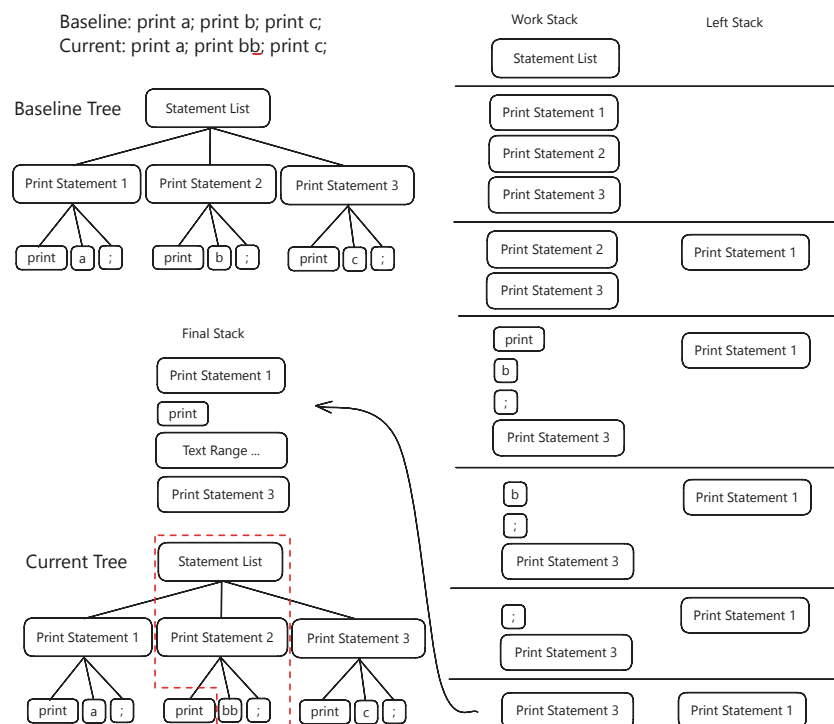
Note that the lexical analyzer, now, does not keep track of position, and can also be called independently of the position, as long as its source is confined to token boundaries. In turn, the incremental parse tree rebuilding algorithm, called the **Blender**, can call the lexer only when it is needed. This also means that the parser cannot use the lexer, doing so would produce brand-new tokens instead of re-using from the previous tree. As such, the Blender and the Lexer implement a common interface, which also enables non-terminal nodes to be exposed to the parser. This interface can be found in Listing 5.

■ **Listing 5** *ITokenStream* interface.

```
internal interface ITokenStream {
    GreenToken PeekToken();
    GreenToken ConsumeToken();
    bool TryPeekNonTerminal([MaybeNullWhen(false)] out GreenNode? n);
    bool TryTakeNonTerminal([MaybeNullWhen(false)] out GreenNode? n);
}
```

The Blender can take the green tree of a previous compilation (the baseline), and locate the change by using a work stack. For this process, the work stack is initialized with the root node of the baseline tree. While the change has not been dealt with, the top of the stack is examined, and if the range overlaps with the change, the node is decomposed into its children. If the top node range is located before the change it is stored into another stack (the left stack), and discarded from the work one. When the top of the stack is a token that falls within the change range, then it is discarded and the algorithm stops. The left stack contains the nodes that fall behind the change, and the work stack the ones that lay after the change. A final stack is produced with these nodes, so that when popped they remain in the order the parser expects, with a caveat the range in the source text change entry is also included in the appropriate place in this stack. Rather than requesting tokens directly, the parser uses an *ITokenStream* interface, that can be found in Listing 5. This enables the parser to operate identically whether it is reading from the lexer (which reads from the source text directly), or the Blender (reading the tokens of the baseline tree and only from the lexer for the changed region). Furthermore, this interface was later extended for the Blender to expose non-terminal nodes, allowing the parser to check in key points (e.g., in a *ParseStatement*) if the next node matches the expected type, avoiding parsing entire sub-trees.

Since in the Blender operations, a token may indeed provoke other tokens to change (e.g., commenting code), the same sorts of interface is needed for the base source and the Blender that the lexer uses, enables following nodes/tokens to be broken up until its textual representation. The *PeekCharacter* that it would contain, cannot modify the current stack, since it may be used by the lexer to detect a token boundary; the same principle applies to the *PeekToken*. The parser can then simply check while parsing if there is a node of the appropriate kind (e.g., a statement) at the beginning of parsing that particular kind, and reusing if that is the case. Figure 3 contains the execution of the Blender in an incremental parser; the added code fragment is highlighted in red, as well as the newly created nodes in the end.



■ **Figure 3** Blender Algorithm.

This algorithm can be trivially extended to support multiple changes. First the list of changes must be in order, and when the first iteration of the specified algorithm finishes, the text change is added to the left stack, and the cycle continues until the end. Furthermore, there is no need to construct a stack at all; one can simply create a tree cursor that knows its position in the tree and is capable of moving to the first child, to the next child (sibling), and to the parent, this way the incremental parse tree rebuilding algorithm can be performed while parsing. This incremental parsing algorithm is based on the algorithm found in the Roslyn compiler, that started as a piece of work of Neal Gafter [20, 21].

7 Conclusion

The initial aim for the project was to build a pedagogical tool that students could use to modify, debug, and self-learn from their tinkering, while empowering them with the background knowledge required to go into production compilers and learn. Given the small

size (<20k lines of code) in comparison to the rich feature set, and the inspiration and similarity to production compilers, we consider that this goal was attained from the design and implementation perspective. However, tests with end users are missing to prove this learning thesis. The incremental parser on its own already pushes the envelopes of what is traditionally thought and discussed in academic environments.

Moreover, the system was built in a modular fashion, following closely the architecture already detailed, which allows for easy usage and swapping of components, enabling other compiler designers to use the platform and extend it without actually modifying it. Furthermore, although the target audience is a master's class on compilers, which is taken after an introductory class of language processing, it also enables other classes besides compiler-centric ones to use this resource as well. Computation systems classes can employ this tool chain to demonstrate the basics (essential concepts) of computer architectures and machine coding in Assembly language. More introductory classes such as language processing can use the intermediate representation, providing students with hands-on experience with LLVM-like infrastructures without the heavy-weight and complexity usually associated with them.

The size of the infrastructure as a whole is showing to be an advantage way beyond the educational sphere. As it currently stands, it is already being used to experiment with new technologies, such as Model Context Protocol built-in to the compiler itself, enabling further research that may one day power the next generation of the programming language ecosystem.

The Lazarus language, while somewhat complex, spanning through all the core concepts of Imperative and Object-Oriented styles, still is a very capable language capable of rather complex “scripts” and graphical user interface capabilities, but fundamentally, it is only intended to provide the means of experimenting with the system and aiding in the research and experimentation taken by students. Though the pedagogical strategy was not discussed in this paper, given it would shift the focus of the paper away from the technical topics and use the necessary space in the process, it is hopefully going to be discussed on a future paper.

Regarding future work, some immediate goals are to continue the development of the webpage and make the entire tool chain generally available. We envision this site, not only as a way to provide the platform as a whole, but also as a means of sharing learning resources, samples and documenting the project with visual and practical explanations to better illustrate the inner workings and the architectural choices made. Furthermore, we want to start conducting diversified testing with end-users to comprehend the learning curve of both using the platform and as a means of a study material. Long-term, the main goal is to evolve the infrastructure to a point where it would be able to self-host itself (i.e., mostly written in the Lazarus language).

References

- 1 Alfred V. Aho, Mónica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, techniques, [and] tools*. Pearson Addison-Wesley, Boston, 2nd. ed edition, 2007. OCLC: 1341171760.
- 2 Anthropic. What is Model Context Protocol? Connect AI to your world, 2025. URL: <https://claude.com/blog/what-is-model-context-protocol>.
- 3 Anthropic. What is the Model Context Protocol (MCP)?, November 2025. URL: <https://modelcontextprotocol.io/docs/getting-started/intro>.
- 4 Andrew W. Appel. *Modern Compiler Implementation in C*. Cambridge University Press, Cambridge, 1997. doi:10.1017/CB09781139174930.
- 5 Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. *Operating Systems: Three Easy Pieces*. Arpaci-Dusseau Books, 1.10 edition, November 2023.

- 6 Thorsten Ball, Nathan Sobo, Antonio Scandurra, and Max Brunsfeld. Rope & SumTree - Zed Blog, 2024. URL: <https://zed.dev/blog/zed-decoded-rope-sumtree>.
- 7 Hans-J. Boehm, Russ Atkinson, and Michael Plass. Ropes: An alternative to strings. *Software: Practice and Experience*, 25(12):1315–1330, 1995. doi:10.1002/spe.4380251203.
- 8 Matthias Braun, Sebastian Buchwald, Sebastian Hack, Roland Leißa, Christoph Mallon, and Andreas Zwinkau. Simple and Efficient Construction of Static Single Assignment Form. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Ranjit Jhala, and Koen De Bosschere, editors, *Compiler Construction*, volume 7791, pages 102–122. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. Series Title: Lecture Notes in Computer Science. doi:10.1007/978-3-642-37051-9_6.
- 9 Max Brunsfeld. Tree Sitter, 2026. doi:10.5281/ZENODO.19357078.
- 10 Cliff Click. Global code motion/global value numbering. In *Proceedings of the ACM SIGPLAN 1995 conference on Programming language design and implementation*, PLDI '95, pages 246–257, New York, NY, USA, June 1995. Association for Computing Machinery. doi:10.1145/207110.207154.
- 11 Cliff Click. From Quads to Graphs: An Intermediate Representation's Journey. Technical report, Rice University, 1997.
- 12 Cliff Click. Simple, 2025. original-date: 2023-08-19T15:52:57Z. URL: <https://github.com/SeaOfNodes/Simple>.
- 13 Cliff Click and Keith D. Cooper. Combining analyses, combining optimizations. *ACM Trans. Program. Lang. Syst.*, 17(2):181–196, 1995. doi:10.1145/201059.201061.
- 14 Cliff Click and Michael Paleczny. A simple graph-based intermediate representation. *SIGPLAN Not.*, 30(3):35–49, 1995. doi:10.1145/202530.202534.
- 15 Clifford Noel Click. *Combining Analysis, Combining Optimizations*. PhD thesis, Rice, 1995. URL: <https://hdl.handle.net/1911/96451>.
- 16 Ron Cytron, Jeanne Ferrante, Barry K. Rosen, Mark N. Wegman, and F. Kenneth Zadeck. Efficiently computing static single assignment form and the control dependence graph. *ACM Transactions on Programming Languages and Systems*, 13(4):451–490, October 1991. doi:10.1145/115372.115320.
- 17 DWARF. DWARF Debugging Information Format Version 5, 2017. URL: <https://dwarfstd.org/doc/DWARF5.pdf>.
- 18 Haoxiang Fei, Yu Zhang, Hongbo Zhang, Yanlin Wang, and Qing Liu. MoonBit: Explore the Design of an AI-Friendly Programming Language. In *Proceedings of the 1st International Workshop on Large Language Models for Code*, LLM4Code '24, pages 79–83, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3643795.3648376.
- 19 Christopher W. Fraser, David R. Hanson, and Todd A. Proebsting. Engineering a simple, efficient code-generator generator. *ACM Lett. Program. Lang. Syst.*, 1(3):213–226, 1992. doi:10.1145/151640.151642.
- 20 Neal Gafter. *Parallel incremental compilation*. PhD thesis, University of Rochester, 1990. URL: <https://urresearch.rochester.edu/institutionalPublicationPublicView.action?institutionalItemId=4743&versionNumber=1>.
- 21 Neal Gafter. Toy Incremental Parser, 2025. URL: <https://github.com/gafter/Toy-Incremental-Parser>.
- 22 Lal George and Andrew W. Appel. Iterated register coalescing. *ACM Trans. Program. Lang. Syst.*, 18(3):300–324, 1996. doi:10.1145/229542.229546.
- 23 Peter Golde, Matthew J. Warren, Neal M. Gafter, and Heejae Chang. Full fidelity parse tree for programming language processing, June 2013. URL: <https://patents.google.com/patent/US20130152061A1/en>.
- 24 Google. Dart and Flutter MCP server, 2026. URL: <https://docs.flutter.dev/ai/mcp-server>.

- 25 R. Hindley. The Principal Type-Scheme of an Object in Combinatory Logic. *Transactions of the American Mathematical Society*, 146:29, December 1969. doi:10.2307/1995158.
- 26 ISO. ISO/IEC 14977:1996: Information technology — Syntactic metalanguage — Extended BNF, 1996. URL: <https://www.iso.org/standard/26153.html>.
- 27 Robert Kennedy, Sun Chan, Shin-Ming Liu, Raymond Lo, Peng Tu, and Fred Chow. Partial redundancy elimination in SSA form. *ACM Trans. Program. Lang. Syst.*, 21(3):627–676, 1999. doi:10.1145/319301.319348.
- 28 Robert Kennedy, Fred Chow, Peter Dahl, Shin-Ming Liu, Raymond Lo, and Mark Streich. Strength reduction via SSAPRE. In Kai Koskimies, Gerhard Goos, Juris Hartmanis, and Jan Van Leeuwen, editors, *Compiler Construction*, volume 1383, pages 144–158. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. doi:10.1007/BFb0026428.
- 29 Oleg Kiselyov, Aggelos Biboudis, Nick Palladinis, and Yannis Smaragdakis. Stream fusion, to completeness. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*, POPL '17, pages 285–299, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3009837.3009880.
- 30 Matthias Koch. Another Look into the Future with Rider's Predictive Debugger | The .NET Tools Blog, 2023. Running Time: 920 Section: .NET Tools. URL: <https://blog.jetbrains.com/dotnet/2023/12/04/another-look-into-the-future-with-riders-predictive-debugger/>.
- 31 Matthias Koch. Introducing Predictive Debugging: A Game-Changing Look into the Future | The .NET Tools Blog, 2023. Section: .NET Tools. URL: <https://blog.jetbrains.com/dotnet/2023/07/27/introducing-predictive-debugging-a-game-changing-look-into-the-future/>.
- 32 C. Lattner and V. Adve. LLVM: A compilation framework for lifelong program analysis & transformation. In *International Symposium on Code Generation and Optimization, 2004. CGO 2004.*, pages 75–86, San Jose, CA, USA, 2004. IEEE. doi:10.1109/CGO.2004.1281665.
- 33 Thomas Lengauer and Robert Endre Tarjan. A fast algorithm for finding dominators in a flowgraph. *ACM Trans. Program. Lang. Syst.*, 1(1):121–141, 1979. doi:10.1145/357062.357071.
- 34 LLVM Foundation. The LLVM Compiler Infrastructure, 2025. original-date: 2016-12-07T09:39:33Z. URL: <https://github.com/llvm/llvm-project>.
- 35 Edward S. Lowry and C. W. Medlock. Object code optimization. *Communications of the ACM*, 12(1):13–22, 1969. doi:10.1145/362835.362838.
- 36 Peng Lyu. Text Buffer Reimplementation, 2018. URL: <https://code.visualstudio.com/blogs/2018/03/23/text-buffer-reimplementation>.
- 37 Microsoft. Roslyn, 2021. URL: <https://github.com/dotnet/roslyn>.
- 38 Microsoft. Language Server Protocol, 2022. URL: <https://microsoft.github.io/language-server-protocol>.
- 39 Microsoft. Debug Adapter Protocol, 2024. URL: <https://microsoft.github.io/debug-adapter-protocol>.
- 40 Robin Milner. A theory of type polymorphism in programming. *Journal of Computer and System Sciences*, 17(3):348–375, December 1978. doi:10.1016/0022-0000(78)90014-4.
- 41 E. Morel and C. Renvoise. Global optimization by suppression of partial redundancies. *Commun. ACM*, 22(2):96–103, 1979. doi:10.1145/359060.359069.
- 42 Noam Nisan. *The elements of computing systems: building a modern computer from first principles*. MIT Press, Cambridge, Massachusetts, 2005.
- 43 Terence Parr. ANTLR, 2024. original-date: 2010-02-04T01:36:28Z. URL: <https://github.com/antlr/antlr4>.
- 44 Reese T. Prosser. Applications of Boolean matrices to the analysis of flow diagrams. In *Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference on - IRE-AIEE-ACM '59 (Eastern)*, pages 133–138, Boston, Massachusetts, 1959. ACM Press. doi:10.1145/1460299.1460314.

- 45 Fabrice Rastello and Florent Bouchez Tichadou, editors. *SSA-based Compiler Design*. Springer International Publishing, Cham, 2022. doi:10.1007/978-3-030-80515-9.
- 46 Michael D. Smith, Norman Ramsey, and Glenn Holloway. A generalized algorithm for graph-coloring register allocation. *SIGPLAN Not.*, 39(6):277–288, 2004. doi:10.1145/996893.996875.
- 47 Vugranam C. Sreedhar and Guang R. Gao. A linear time algorithm for placing phi-nodes. In *Proceedings of the 22nd ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '95, pages 62–73, New York, NY, USA, 1995. Association for Computing Machinery. doi:10.1145/199448.199464.
- 48 Thomas VanDrunen and Antony L. Hosking. Value-Based Partial Redundancy Elimination. In Gerhard Goos, Juris Hartmanis, Jan Van Leeuwen, and Evelyn Duesterwald, editors, *Compiler Construction*, volume 2985, pages 167–184. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. Series Title: Lecture Notes in Computer Science. doi:10.1007/978-3-540-24723-4_12.
- 49 Matthew J. Warren, Avner Y. Aharoni, Mads Torgersen, Renaud Paquay, Neal M. Gafter, Jared Parsons, David N. Schach, Aleksey V. Tsingauz, Peter Golde, Kevin Pilch-Bisson, and Karen Liu. Efficient immutable syntax representation with incremental change, February 2020. URL: <https://patents.google.com/patent/US10564944/en>.
- 50 Matthew J. Warren, Mads Torgersen, Renaud Paquay, Neal M. Gafter, Jared Parsons, David N. Schach, Aleksey V. Tsingauz, Peter Golde, Kevin Andrew Pilch, and Karen Liu. Efficient immutable syntax representation with incremental change, June 2022. URL: <https://patents.google.com/patent/US11372630B2/en>.
- 51 Mark N. Wegman and F. Kenneth Zadeck. Constant propagation with conditional branches. *ACM Trans. Program. Lang. Syst.*, 13(2):181–210, 1991. doi:10.1145/103135.103136.

Out of the Loop No More: Online Abstract Debugging with Loop Stepping

João Alvim  

University of Minho, Braga, Portugal

Maarten Steevens  

Ghent University, Belgium

Christophe Scholliers  

Ghent University, Belgium

Abstract

Traditional debugging makes it practically impossible to manually explore all execution paths, often leaving hypotheses insufficiently explored. Abstract debugging addresses this problem by safely over-approximating all program behaviours through a familiar stepping interface. However, existing approaches operate *offline*, pre-computing fixed loop summaries that developers cannot interactively refine or inspect at the individual iteration level.

In this paper, we propose *online abstract debugging*, which performs abstract interpretation on demand as the developer steps through the program. This novel architecture enables *loop stepping*: developers can step into a loop body for precise, iteration-specific states, or step-over to obtain a coarser summary. By interleaving analysis with execution, developers gain direct control over the cost-precision trade-off, computing high precision only where needed to settle a hypothesis. We instantiate this model for WebAssembly and present a prototype implementation, demonstrating that developers can effectively draw sound conclusions about complex, potentially non-terminating looping programs.

2012 ACM Subject Classification Software and its engineering → Software testing and debugging

Keywords and phrases Debugging, Abstract Interpretation

Digital Object Identifier 10.4230/OASICS.SLATE.2026.20

Category Short Paper

Supplementary Material

Software (Prototype): <https://live-abstract-debugger.surge.sh>

Funding *Maarten Steevens*: Funded by the Research Foundation Flanders, grant number 1SA6C26N.

1 Introduction

Debugging is a highly time-consuming process [1], but traditional debuggers are limited to single concrete executions, making it laborious to thoroughly test loops and explore all possible program behaviours [20]. Abstract debugging [9] addresses this by allowing developers to step through abstract states that safely over-approximate all executions simultaneously using a familiar stepping interface. However, existing abstract debuggers operate *offline*: a static analyser pre-computes fixed summaries for the entire program before the debugging session begins. Consequently, if a loop summary loses precision, the developer cannot interactively explore individual iterations on the fly, they are forced to either accept the coarse results or restart the entire analysis (which needs to run to completion) with adjusted global parameters, such as increased loop unrolling. Overcoming these opaque workflows by focusing on usability and explainability is crucial, as poor developer experience frequently drives users to abandon static analysis tools [5].



© João Alvim, Maarten Steevens, and Christophe Scholliers;
licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 20; pp. 20:1–20:10

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

To overcome this, we propose *online abstract debugging*, where abstract interpretation is performed on demand as the user steps through the program. This architectural shift directly enables *loop stepping*. Developers can step into a loop to explore precise, iteration-specific abstract states, or step-over to obtain coarser loop summaries. By interleaving analysis with interaction, our approach grants the developer interactive control over the cost-precision trade-off, computing high precision only where a hypothesis demands it, without committing to a full analysis of all iterations and while preserving soundness throughout.

In this work in progress paper we detail how such an abstract debugger can be build formally and report on an initial implementation of this model for WebAssembly [8], a low-level stack-based compilation target widely used as a backend for languages [19] as diverse as Python, Go, and Haskell. We make the following contributions:

- We propose online abstract debugging and identify loop stepping as its principal consequence, extending the abstract debugging model of [9] with interactive loop unrolling.
- We present a *recipe* for systematically defining a live abstract debugger starting from a small-step operational semantics, which we illustrate using a simple WHILE language.
- We report on an initial implementation of our technique for debugging WebAssembly programs.

2 Online Abstract Debugging by Example

Before diving into the formal details, we first illustrate the core concepts of our online abstract debugger through a motivating example ¹. When using a traditional debugger, developers are limited to inspecting the state of a single, specific execution path, leaving them blind to how the program might behave across all possible executions. Imagine opening a standard interactive debugger, but instead of the variables pane showing exact values like $x = 4$, it displays abstractions of the concrete state, for example *intervals* representing all possible values a variable might take, such as $x \in [0, 10]$.

It allows developers to interactively step through all possible program executions at once by means of abstract interpretation [7]. To guarantee that the analysis does not get stuck in infinite loops, abstract interpretation engines use a technique called *widening*. If a variable's bounds keep growing during a loop, widening aggressively assumes the bounds might grow forever, pushing them to infinity (e.g. $+\infty$). While this guarantees the analyzer finishes, it often results in a loss of precision for the user.

To illustrate our debugger, consider the following program in the WHILE language. A developer reads an unknown integer z from the user, then runs a short loop that overwrites z with the constant 4 and increments a counter x :

```
input(z) ; x := 0 ; while(x < 2){ z := 4 ; x := x + 1 }
```

Starting from an unknown value of z and $x = 0$, the loop runs exactly twice (x passes through 1 and then 2), assigning $z := 4$ on every iteration. Although this program is trivially simple, let us pretend that a developer is actively debugging it to verify its behavior.

Exploring abstract states

Suppose the developer places a breakpoint at the loop head. In the variables pane they see $x \in [0, 0]$ and $z \in [-\infty, +\infty]$. They want to quickly skip past the loop, so they click the **Step Over** button.

¹ Try out this example here: <https://live-abstract-debugger.surge.sh>

Behind the scenes, the debugger hands control to the automated analyzer to summarize the entire loop. The analyzer joins the pre-loop state $x \in [0, 0]$ with the post-body state $x \in [1, 1]$ at the loop head, obtaining $x \in [0, 1]$. Fearing an infinite loop, widening kicks in, pushing the assumed upper bound to $+\infty$: the loop-head invariant becomes $x \in [0, +\infty]$. At the same loop head, $z \in [-\infty, +\infty]$ (the unknown input value) is joined with $z \in [4, 4]$ (the post-body value), yielding $z \in [-\infty, +\infty]$. When the loop terminates, the debugger applies the exit condition $x \geq 2$.

The execution marker jumps to the end of the program, and the user looks at the variables pane. It now reads $x \in [2, +\infty]$ and $z \in [-\infty, +\infty]$. The analysis is perfectly safe and mathematically sound, but the information that the loop always sets z to exactly 4 has been completely lost.

Gaining precision by manual unrolling

Realizing that $z \in [-\infty, +\infty]$ is too imprecise to be useful, the developer restarts the debugger. This time, arriving at the loop head with $x \in [0, 0]$ and $z \in [-\infty, +\infty]$, they decide to guide the analyzer manually by clicking **Step Into**.

By manually stepping through the code, the user forces the debugger to evaluate exact bounds step-by-step, preventing the automated widening heuristic from triggering. Because $x \in [0, 0]$ is a singleton interval, the guard $x < 2$ is unambiguously true and the assignment $z := 4$ executes without any branching. The user watches the variables pane update in real-time: after the first iteration, the state becomes $x \in [1, 1]$ with $z \in [4, 4]$, and after the second iteration, it updates to $x \in [2, 2]$ with $z \in [4, 4]$.

At this point the developer steps once more to evaluate the loop guard ($x < 2$). Because the debugger knows exactly that $x \in [2, 2]$, it determines that the loop condition is unambiguously false. The debugger exits the loop naturally.

The final state in the variables pane is exactly $x \in [2, 2]$ and $z \in [4, 4]$. By simply interacting with the debugger as they would in any standard IDE, the developer acts as a human loop unroller and successfully bypasses the precision loss of widening.

This highlights the core novelty of our approach: *online abstract debugging*. Whereas previous abstract debuggers operate offline our tool performs abstract interpretation on demand over a live program. This enables *loop stepping*, granting the developer direct, interactive control over the cost-precision trade-off.

3 Recipe: Defining an Online Abstract Debugger

In this section, following the approach to abstract interpretation outlined in [15], we present an overview of how to construct an online abstract debugger for a small imperative programming language: WHILE. We build the formal model of the debugger in three stages: first, we establish the *collecting* small-step operational semantics of the WHILE language. Next, we lift this execution model to define an abstract interpreter and finally, we formalize our interactive debugger semantics on top of this abstract semantics.

3.1 The while Language

Given the intuition for our online abstract debugger, we proceed to formalize its underlying mechanics. We use the WHILE language, a small didactical imperative language, to illustrate the concept of online abstract debugging. The syntax of the WHILE language is shown in Figure 1.

$$\begin{aligned}
C &::= \mathbf{skip} \mid C; C \mid x := E \mid \mathbf{input}(x) \mid \mathbf{if}(B)\{C\}\mathbf{else}\{C\} \mid \mathbf{while}(B)\{C\} \\
E &::= n \mid x \mid E + E \\
B &::= \mathbf{true} \mid \mathbf{false} \mid E < E \mid E = E \\
P &::= C
\end{aligned}$$

■ **Figure 1** Syntax of the WHILE language.

$$\begin{aligned}
&\frac{}{\langle x := E, m \rangle \rightarrow \langle \mathbf{skip}, m[x \mapsto eval_E(m)] \rangle} \text{ (E-ASSIGN)} \\
&\frac{z \text{ is an input integer}}{\langle \mathbf{input}(x), m \rangle \rightarrow \langle \mathbf{skip}, m[x \mapsto z] \rangle} \text{ (E-INPUT)} \quad \frac{\langle C_1, m \rangle \rightarrow \langle C'_1, m' \rangle}{\langle C_1; C_2, m \rangle \rightarrow \langle C'_1; C_2, m' \rangle} \text{ (E-SEQ-STEP)} \\
&\frac{}{\langle \mathbf{skip}; C_2, m \rangle \rightarrow \langle C_2, m \rangle} \text{ (E-SEQ-FINISH)} \quad \frac{eval_B(m) = \mathbf{true}}{\langle \mathbf{if}(B)\{C_1\}\mathbf{else}\{C_2\}, m \rangle \rightarrow \langle C_1, m \rangle} \text{ (E-IF-TRUE)} \\
&\frac{eval_B(m) = \mathbf{false}}{\langle \mathbf{if}(B)\{C_1\}\mathbf{else}\{C_2\}, m \rangle \rightarrow \langle C_2, m \rangle} \text{ (E-IF-FALSE)} \\
&\frac{}{\langle \mathbf{while}(B)\{C\}, m \rangle \rightarrow \langle \mathbf{if}(B)\{C; \mathbf{while}(B)\{C\}\}\mathbf{else}\{\mathbf{skip}\}, m \rangle} \text{ (E-WHILE)}
\end{aligned}$$

■ **Figure 2** Small-Step Operational Semantics of the WHILE Language.

We differentiate between commands and expressions. Commands (C) are state-modifying statements that dictate the control flow of the program and update the environment, whereas expressions evaluate to specific values without producing side effects. Commands (C) consist of standard constructs including no-operations (**skip**), sequences ($C; C$), assignments ($x := E$), inputs (**input**(x)), conditionals (**if**), and loops (**while**). Arithmetic expressions (E) and boolean expressions (B) are standard. Finally, a complete program (P) is defined as a single top-level command C .

Step1: Small-Step Operational Semantics

We define the small-step operational semantics of the WHILE language in figure 2. The evaluation relation $\langle C, m \rangle \rightarrow \langle C', m' \rangle$ transitions a configuration of a command C and memory state m by a single computational step, terminating at the **skip** command. For brevity, expression evaluations ($eval_E$ and $eval_B$) are abstracted as deterministic, side-effect-free functions.

The rules operate as follows: (E-ASSIGN) and (E-INPUT) update the memory m with an evaluated expression or external input z , respectively, reducing to **skip**. Sequential composition (E-SEQ-STEP, E-SEQ-FINISH) evaluates the left command until it reduces to **skip**, then advances to the right. Conditionals (E-IF-TRUE, E-IF-FALSE) perform standard branching based on the boolean guard. Finally, (E-WHILE) defines loop execution via unrolling, expanding into an **if** command that executes the body and repeats, or terminates.

Step2: Abstract Semantics

Given the small-step operational semantics of our language, we can proceed to define the abstract semantics. We establish this by lifting our evaluation to operate over an abstract memory $M^\sharp$ (for example, a memory of intervals) rather than a concrete memory m . The abstract transition relation $\langle C, M^\sharp \rangle \rightarrow^\sharp \langle C', M'^\sharp \rangle$ maps a configuration to a valid successor configuration.

The rules adapt their concrete counterparts to the abstract domain: (A-ASSIGN) evaluates expressions using an abstract evaluation function $eval_E^\sharp$ and modifies the state via $update^\sharp$. Because user input is entirely unknown, (A-INPUT) safely over-approximates it by assigning the top element $\top$. Sequential composition (A-SEQ-STEP, A-SEQ-FINISH) propagates the transition directly.

For branching, note that both (A-IF-TRUE) and (A-IF-FALSE) might apply simultaneously if the abstract state cannot definitively prove the condition true or false. For example, the condition $x > 0$ given $x \in [-5, 5]$ can evaluate to both true and false. To handle this, these rules rely on a $filter^\sharp$ function that effectively splits the abstract memory into two disjoint parts: one encompassing the states that satisfy the condition, and another for those that do not. A transition along a given branch is only valid if its corresponding filtered memory remains feasible (i.e., $\neq \perp$).

Finally, (A-WHILE) handles loops via unrolling.

$$\begin{array}{c}
 \textbf{Abstract Evaluation} \qquad \qquad \qquad \langle C, M^\sharp \rangle \rightarrow^\sharp \langle C', M'^\sharp \rangle \\
 \\
 \frac{}{\langle x := E, M^\sharp \rangle \rightarrow^\sharp \langle \text{skip}, update_x^\sharp(M^\sharp, eval_E^\sharp(M^\sharp)) \rangle} \text{ (A-ASSIGN)} \\
 \\
 \frac{}{\langle \text{input}(x), M^\sharp \rangle \rightarrow^\sharp \langle \text{skip}, update_x^\sharp(M^\sharp, \top) \rangle} \text{ (A-INPUT)} \\
 \\
 \frac{\langle C_1, M^\sharp \rangle \rightarrow^\sharp \langle C'_1, M'^\sharp \rangle}{\langle C_1; C_2, M^\sharp \rangle \rightarrow^\sharp \langle C'_1; C_2, M'^\sharp \rangle} \text{ (A-SEQ-STEP)} \qquad \frac{}{\langle \text{skip}; C_2, M^\sharp \rangle \rightarrow^\sharp \langle C_2, M^\sharp \rangle} \text{ (A-SEQ-FINISH)} \\
 \\
 \frac{filter_B^\sharp(M^\sharp) \neq \perp}{\langle \text{if}(B)\{C_1\}\text{else}\{C_2\}, M^\sharp \rangle \rightarrow^\sharp \langle C_1, filter_B^\sharp(M^\sharp) \rangle} \text{ (A-IF-TRUE)} \\
 \\
 \frac{filter_{\neg B}^\sharp(M^\sharp) \neq \perp}{\langle \text{if}(B)\{C_1\}\text{else}\{C_2\}, M^\sharp \rangle \rightarrow^\sharp \langle C_2, filter_{\neg B}^\sharp(M^\sharp) \rangle} \text{ (A-IF-FALSE)} \\
 \\
 \frac{}{\langle \text{while}(B)\{C\}, M^\sharp \rangle \rightarrow^\sharp \langle \text{if}(B)\{C; \text{while}(B)\{C\}\}\text{else}\{\text{skip}\}, M^\sharp \rangle} \text{ (A-WHILE)}
 \end{array}$$

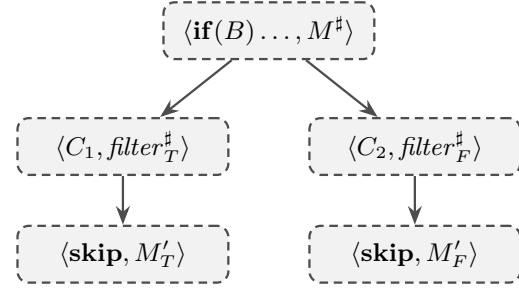
■ **Figure 3** Abstract operational semantics of the WHILE language.

Abstract Step Function and Analysis. While the abstract semantics describes how a single configuration evolves, a static analysis must track the evolution of the entire program state. We define an *abstract step function* $Step^\sharp(\mathcal{S}^\sharp)$ that propagates a set of abstract configurations $\mathcal{S}^\sharp$. This function collects all possible successors ($\langle C, M^\sharp \rangle \rightarrow^\sharp$), for a single command C , as visualized in Figure 4 (Right) and joins the abstract memories of configurations, i.e. M'_T and M'_F .

Algorithm: FIX

```

1:  $\mathcal{S}^\# \leftarrow \{\langle P, \perp \rangle\}$ 
2: repeat
3:    $\mathcal{R}^\# \leftarrow \mathcal{S}^\#$ 
4:    $\mathcal{S}^\# \leftarrow \mathcal{R}^\# \cup^\nabla \text{Step}^\#(\mathcal{R}^\#)$ 
5: until  $\mathcal{S}^\# \sqsubseteq \mathcal{R}^\#$ 
6: return  $\mathcal{S}^\#$ 
    
```



■ **Figure 4 (Left)** The automated fixpoint analysis algorithm using widening ($\cup^\nabla$). **(Right)** Visualizing abstract branching where multiple feasible paths create separate frontiers.

To perform a complete analysis, we compute the least fixpoint of this step function as shown in Figure 4 (Left). However, because our interval domain has infinite height, a standard iteration might never stabilize. We therefore use a *widening operator* (∇) to force convergence by aggressively pushing growing bounds to infinity. For an interval $[lo, hi]$, widening is defined as:

$$[lo_1, hi_1] \nabla [lo_2, hi_2] = [(lo_2 < lo_1 ? -\infty : lo_1), (hi_2 > hi_1 ? +\infty : hi_1)]$$

This ensures that any ascending chain of abstract memories stabilizes in a finite number of steps. The complete analysis algorithm, which computes a stable set of abstract configurations $\mathcal{S}^\#$, is shown in Figure 4.

The widened union $\mathcal{A} \cup^\nabla \mathcal{B}$ merges successors into the current state. If a bound at a specific program point grows, the operator applies ∇ to jump to a stable limit. This algorithm is what powers the STEPOVER command in our debugger: it automatically computes the entire loop summary, potentially sacrificing precision for speed.

Step3: Online Abstract Debugger Semantics

The final step is to define the abstract debugger semantics. Rather than passively analyzing the code, these semantics must accommodate interactive commands and manage the multiple states resulting from abstract branching. The key novelty of this semantics is that it allows the user to navigate divergent branches and dynamically tune analysis precision.

Debugger Configuration. A debugger configuration is a tuple $\langle op, C, M^\#, \sigma \rangle$, where $op \in \{\text{Step}, \text{StepOver}, \text{StepOut}\}$ is the user's current debug operation, $C \in \text{Cmd}$ and $M^\# \in \mathbb{M}^\#$ form the current abstract configuration, and σ is an execution stack used to manage branching.

When an abstract condition cannot be definitively proven true or false, the execution must split. The stack σ enables the debugger to explore one branch while suspending alternative paths. Each stack frame $(pending, acc)$ tracks a list of unexplored configurations (*pending*) alongside an accumulated memory (*acc*) that joins the abstract states of all branches that have completed execution at the current level.

Debugger Transition Rules. The debugger transition relation $\rightarrow_{\text{dbg}}$ evaluates a given operation applied to the current state, mapping it to the resulting execution state. It is defined by the five rules in Figure 5.

$$\begin{array}{c}
\text{Debugger Transition} \qquad \langle op, C, M^\sharp, \sigma \rangle \rightarrow_{\text{dbg}} \langle C', M'^\sharp, \sigma' \rangle \\
\\
\frac{\text{unique } \langle C, M^\sharp \rangle \rightarrow^\sharp \langle C', M'^\sharp \rangle}{\langle \text{Step}, C, M^\sharp, \sigma \rangle \rightarrow_{\text{dbg}} \langle C', M'^\sharp, \sigma \rangle} \text{ (D-STEP)} \quad \frac{}{\langle \text{Step}, \text{skip}, M^\sharp, \sigma \rangle \rightarrow_{\text{dbg}} \text{resume}(M^\sharp, \sigma)} \text{ (D-FINISH)} \\
\\
\frac{\langle C, M^\sharp \rangle \rightarrow^\sharp \langle C_1, M_1^\sharp \rangle \quad \langle C, M^\sharp \rangle \rightarrow^\sharp \langle C_2, M_2^\sharp \rangle \quad C_1 \neq C_2}{\langle \text{Step}, C, M^\sharp, \sigma \rangle \rightarrow_{\text{dbg}} \langle C_1, M_1^\sharp, ([\langle C_2, M_2^\sharp \rangle], \perp) :: \sigma \rangle} \text{ (D-BRANCH)} \\
\\
\frac{}{\langle \text{StepOver}, \text{while}(B)\{C\}, M^\sharp, \sigma \rangle \rightarrow_{\text{dbg}} \text{resume}(\text{fix}(\text{while}(B)\{C\}, M^\sharp), \sigma)} \text{ (D-STEP-OVER)} \\
\\
\frac{}{\langle \text{StepOut}, C, M^\sharp, (\text{pending}, \text{acc}) :: \sigma \rangle \rightarrow_{\text{dbg}} \text{resume}(\text{exhaust}(C, M^\sharp, \text{pending}, \text{acc}), \sigma)} \text{ (D-STEP-OUT)}
\end{array}$$

■ **Figure 5** The live debugger operational semantics.

The (D-STEP) rule handles standard, deterministic execution where the abstract memory yields a single valid successor. Conversely, (D-BRANCH) applies when the abstract memory is consistent with both outcomes of a branch condition, yielding two valid transitions. The debugger immediately steps into one branch (C_1) while pushing the other (C_2) onto the stack σ . The (D-STEP-OVER) rule applies exclusively when the current command is a **while**-loop and the user chooses to skip it. It invokes the *fix* helper (as described in Section 2) to instantly compute the summary, entirely bypassing manual iteration. The (D-STEP-OUT) rule runs C and all pending siblings to completion with *exhaust* and calls *resume* on the parent stack, popping exactly one frame. Finally, (D-FINISH) intercepts any execution that reaches **skip** and triggers the *resume* routine to unwind the stack. The *resume* helper determines the next state by inspecting the current stack σ :

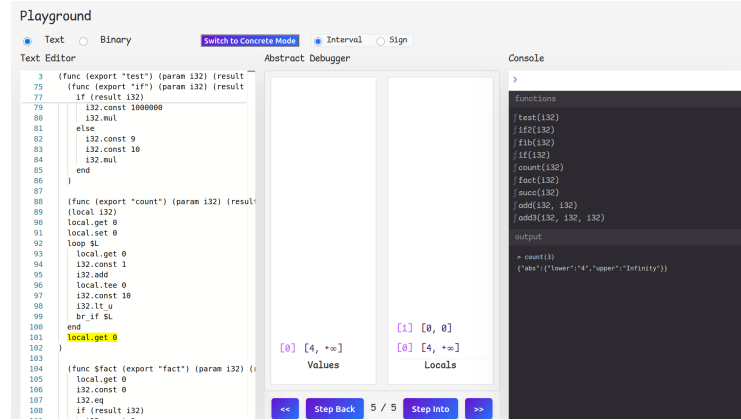
$$\text{resume}(M^\sharp, \sigma) = \begin{cases} \langle \text{skip}, M^\sharp, [] \rangle & \text{if } \sigma = [] \\ \langle C', M'^\sharp, (\text{rest}, \text{acc} \sqcup M^\sharp) :: \sigma' \rangle & \text{if } \sigma = (\langle C', M'^\sharp \rangle :: \text{rest}, \text{acc}) :: \sigma' \\ \text{resume}(\text{acc} \sqcup M^\sharp, \sigma') & \text{if } \sigma = ([], \text{acc}) :: \sigma' \end{cases}$$

The first case signifies total termination. The second case pops the next pending branch to explore it, joining the just-finished memory $M^\sharp$ into the frame's accumulator. The third case handles an exhausted frame by joining its accumulator upward into the parent frame and recursing.

4 An Online Abstract Debugger for WebAssembly

We implemented the formalisation presented in the previous sections as both a concrete and abstract debugger for Miniwasm [17]. Built in TypeScript atop the `wasm-ts-runtime` engine², our tool supports parsing, executing, and abstract debugging both WebAssembly text files and raw binaries. The web-based interface (Figure 6) consists of three primary sections: a text editor (left) highlighting the active instruction, a state viewer (middle) dynamically displaying the abstract operand stack and local variables (e.g., intervals like $[4, +\infty]$), and an interactive console (right). A top configuration bar enables users to toggle

² <https://github.com/ryohey/ts-wasm-runtime>



■ **Figure 6** The Miniwasm web interface operating in Abstract Debugger mode.

between concrete and abstract modes, select abstract domains (such as Interval or Sign), and switch input formats. Execution flow is managed via a bottom control bar, providing standard operations to step backward, step into, or advance through the program’s abstract frontiers.

5 Related Work

The term *abstract debugging* [3] was first coined by Bourdoncle to describe an offline static analysis driven by user-defined predicates manually annotated in the source code. This is quite different from our approach, where we define an *online* abstract debugger instead of a static analysis framework. Similarly, program slicing focuses on extracting only the statements relevant to a slicing criterion [11], whereas abstract debugging focuses on inspecting sound over-approximated program states during execution.

Building on the idea of abstract debugging, Holter et al. [9] use an inter-procedural Abstract Reachability Graph (iARG) to map static threads analysis results to standard debugger interfaces. While this enables the exploration of feasible program paths, the underlying static analyzer must still compute all fixpoints and apply loop widening before the interactive session begins. Similarly, other approaches provide comprehensive offline reports [4] or interactive, web-based interfaces like Try-Mopsa [13]. Additionally, some WebAssembly tools, such as Wassail [18], WasmGrind [12], and Wastrumentation [14], have been developed, but their core analysis remains strictly offline.

Alternatively, random property-based testing [6], state space exploration techniques such as model checking used in [2], symbolic execution [10], and multiverse debugging [16] allow for interactive navigation but struggle with state and path explosion. Abstract debugging inherently mitigates this explosion by operating over abstract domains. Building on this advantage, we propose *online abstract debugging*, which replaces offline precomputation with abstract states dynamically computed on demand. This allows developers to interactively explore intermediate states, such as those inside loops, before widening occurs.

6 Conclusion

In this paper, we introduced *online abstract debugging*. By interleaving abstract interpretation directly with the debugging session, our architecture enables *loop stepping*. This grants developers interactive control over the cost-precision trade-off, allowing them to manually

unroll loops for iteration-specific precision or step over them using automated summaries. We formalized this approach systematically, lifting a standard small-step operational semantics to a live debugger, and demonstrated its practical viability through a web-based prototype for WebAssembly (Miniwasm). Future work will expand our model to support full WebAssembly features, integrate relational abstract domains, and evaluate the approach through an empirical user study.

References

- 1 Abdulaziz Alaboudi and Thomas D. LaToza. An exploratory study of debugging episodes, 2021. [arXiv:2105.02162](#).
- 2 Johan Bengtsson, Kim G. Larsen, Fredrik Larsson, Paul Pettersson, and Wang Yi. UPPAAL – A Tool Suite for Automatic Verification of Real-Time Systems. *BRICS Report Series*, 3(58), 1996. doi:10.7146/brics.v3i58.18769.
- 3 François Bourdoncle. Abstract debugging of higher-order imperative languages. *SIGPLAN Not.*, 28(6):46–55, 1993. doi:10.1145/173262.155095.
- 4 David Bühler, André Maroneze, and Valentin Perrelle. Abstract interpretation with the EvaEva (Frama-C plug-in) plug-in. In Nikolai Kosmatov, Virgile Prevosto, and Julien Signoles, editors, *Guide to Software Verification with Frama-C: Core Components, Usages, and Applications*, pages 131–186. Springer International Publishing, Cham, 2024. doi:10.1007/978-3-031-55608-1_3.
- 5 Maria Christakis and Christian Bird. What developers want and need from program analysis: an empirical study. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, ASE '16*, pages 332–343, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2970276.2970347.
- 6 Koen Claessen and John Hughes. Quickcheck: a lightweight tool for random testing of haskell programs. *SIGPLAN Not.*, 35(9):268–279, 2000. doi:10.1145/357766.351266.
- 7 Patrick Cousot and Radhia Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the 4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages, POPL '77*, pages 238–252, New York, NY, USA, 1977. Association for Computing Machinery. doi:10.1145/512950.512973.
- 8 Andreas Haas, Andreas Rossberg, Derek L. Schuff, Ben L. Titzer, Michael Holman, Dan Gohman, Luke Wagner, Alon Zakai, and JF Bastien. Bringing the web up to speed with WebAssembly. In *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2017*, pages 185–200, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3062341.3062363.
- 9 Karoline Holter, Juhan Oskar Hennoste, Patrick Lam, Simmo Saan, and Vesal Vojdani. Abstract debuggers: Exploring program behaviors using static analysis results. In *Proceedings of the 2024 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, Onward! '24*, pages 130–146, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3689492.3690053.
- 10 Hao Jin, Yanyan Jiang, Na Liu, Chang Xu, Xiaoxing Ma, and Jian Lu. Concolic metamorphic debugging. In *2015 IEEE 39th Annual Computer Software and Applications Conference*, volume 2, pages 232–241, 2015. doi:10.1109/COMPSAC.2015.79.
- 11 Isabella Mastroeni and Damiano Zanardini. Abstract program slicing: An abstract interpretation-based approach to program slicing. *ACM Trans. Comput. Logic*, 18(1), 2017. doi:10.1145/3029052.
- 12 Peter Thiemann Michael Staab. Wasmgrind: Towards dynamic concurrency analysis for multithreaded webassembly. In *Workshop proceedings (metadata pending)*, 2025.
- 13 Raphaël Monat. Try-mopsa: Relational static analysis in your pocket. In Yu-Fang Chen, Thomas Jensen, and Ondřej Lengál, editors, *Verification, Model Checking, and Abstract Interpretation*, pages 197–211, Cham, 2026. Springer Nature Switzerland. doi:10.1007/978-3-032-15700-3_10.

- 14 Aäron Munsters, Angel Luis Scull Pupo, and Elisa Gonzalez Boix. Wastrumentation: Portable WebAssembly Dynamic Analysis with Support for Intercession. In Jonathan Aldrich and Alexandra Silva, editors, *39th European Conference on Object-Oriented Programming (ECOOP 2025)*, volume 333 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:29, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ECOOP.2025.23.
- 15 Xavier Rival and Kwangkeun Yi. *Introduction to Static Analysis: An Abstract Interpretation Perspective*. MIT Press, 2020.
- 16 Maarten Steevens, Tom Lauwaerts, and Christophe Scholliers. Concolic multiverse debugging. In *Proceedings of the 2nd ACM International Workshop on Future Debugging Techniques, DEBT 2024*, pages 28–29, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3678720.3685318.
- 17 Quentin Stiévenart and Coen De Roover. Compositional information flow analysis for webassembly programs. In *2020 IEEE 20th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, pages 13–24, 2020. doi:10.1109/SCAM51674.2020.00007.
- 18 Quentin Stiévenart and Coen De Roover. Wassail: a webassembly static analysis library, 2021.
- 19 WebAssembly Community Group. Developer’s guide – WebAssembly. <https://webassembly.org/getting-started/developers-guide/>. Acedido a: 18 abr. 2026.
- 20 Andreas Zeller. *Why Programs Fail: A Guide to Systematic Debugging*. Elsevier/Morgan Kaufmann, Amsterdam; Boston, 2nd edition, 2009.

RI: A DSL for Software Development Project Continuous Configuration

Cássio Ritse Machado Dos Santos Silva ✉🏠

Instituto Politécnico de Bragança, Portugal

Maria João Varanda Pereira ✉🏠^{ID}

Research Centre in Digitalization and Intelligent Robotics (CeDRI),
Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC),
Instituto Politécnico de Bragança, Portugal

Paulo Alexandre Vara Alves ✉🏠^{ID}

Research Centre in Digitalization and Intelligent Robotics (CeDRI),
Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC),
Instituto Politécnico de Bragança, Portugal

Gustavo Jansen de Souza Santos ✉🏠^{ID}

Universidade Tecnológica Federal do Paraná (UTFPR), Dois Vizinhos, Brazil

Abstract

This paper addresses the challenges related to standardization and automation in the development and maintenance of microservice-based architectures, particularly in heterogeneous environments involving multiple development teams. In such contexts, the absence of unified standards for service definition and implementation, combined with technological diversity and team fragmentation, can lead to architectural inconsistencies, increased integration effort, difficulties in system evolution, and higher maintenance costs over time. Additionally, a significant portion of configuration, orchestration, and artifact generation tasks is performed manually, increasing the risk of errors and rework. As a solution, this paper proposes a Domain-Specific Language (DSL), named RI language, designed for the declarative description of microservices, along with a supporting tool, TSE (Toolbox Service Executor), implemented in Python (PLY) and based on the RI grammar. The tool is responsible for interpreting specifications written in RI and automating the generation of software artifacts through extensible plugins. The proposed approach enables the representation of structural and operational aspects of services in a technology-agnostic manner, promoting consistency, reuse, and reduction of manual effort.

2012 ACM Subject Classification Software and its engineering → Domain specific languages

Keywords and phrases DSL, microservices, software configuration, code generator

Digital Object Identifier 10.4230/OASICS.SLATE.2026.21

Funding This work was supported by FCT – Fundação para a Ciência e Tecnologia, I.P. by projects: CeDRI, UID/05757/2025 (DOI: <https://doi.org/10.54499/UID/05757/2025>) and UID/-PRR/05757/2025 (DOI: <https://doi.org/10.54499/UID/PRR/05757/2025>); SusTEC, LA/P/0007/2020 (DOI: <https://doi.org/10.54499/LA/P/0007/2020>).

Acknowledgements The authors would also like to thank the supervising professors and co-authors for their guidance and collaboration, as well as the Universidade Tecnológica Federal do Paraná (UTFPR), Brazil, and the Instituto Politécnico de Bragança (IPB), Portugal, for their institutional support.

1 Introduction

Microservice architecture has been widely adopted in the development of modern distributed systems, primarily due to its ability to promote scalability, modularity, and independence among components. However, as the number of services increases, so does the complexity associated with standardization, integration, and maintenance of these systems [9].



© Cássio Ritse Machado Dos Santos Silva, Maria João Varanda Pereira, Paulo Alexandre Vara Alves, and Gustavo Jansen de Souza Santos;

licensed under Creative Commons License CC-BY 4.0

15th Symposium on Languages, Applications and Technologies (SLATE 2026).

Editors: Fernando Batista, Eugénio Ribeiro, Ricardo Ribeiro, and André L. Santos; Article No. 21; pp. 21:1–21:14

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

One of the main challenges in this context is ensuring consistency in how different microservices are structured and exposed, especially in environments involving multiple teams and technologies. Even when different teams use the same programming language and similar frameworks, it is common for them to adopt distinct implementation patterns for aspects such as code organization, route definitions, service configuration, and communication strategies [6]. These variations, although often subtle, can lead to significant inconsistencies, making system maintenance, evolution, and service integration more difficult. For example, different teams may adopt REST (Representational State Transfer) conventions with explicit versioning in the URL (e.g., `/v1/users`), while others may omit versioning or use different naming and routing structures. Additionally, differences in response formats may require specific adaptations on the consumer side, increasing coupling and integration complexity.

Although each microservice maintains autonomy over its business logic implementation, aspects such as communication, configuration, structural organization, and implementation patterns often require standardization [6]. The absence of clear guidelines for these elements can result in rework and increased maintenance effort.

Given this scenario, it becomes relevant to define mechanisms that allow the standardized and technology-agnostic description of structural and operational characteristics of software projects. In this context, solutions based on Domain-Specific Languages (DSLs) and language processing techniques emerge as a promising approach, as they enable the declarative modeling of domain-specific elements [1].

In this work, we propose the RI DSL for the description of software projects, along with the development of a toolbox, TSE, which enables the interpretation of these specifications and the association of semantic actions to automate the generation and configuration of software artifacts. The goal is to increase automation in the integration and configuration of software projects across different teams and technological contexts, in a consistent and standardized manner.

2 Background

A microservice can be defined as an autonomous software component, independently deployed and maintained, whose purpose is to execute complete business logic or perform specific tasks. These services are organized within a microservice-based architecture, in which each component contributes to a broader objective, collectively forming a complete distributed system [4].

One of the fundamental principles of this architecture is the use of standardized service contracts, through which microservices expose their functionalities, capabilities, and interaction policies. These contracts play a crucial role in service communication, ensuring interoperability and loose coupling among system components [11].

Therefore, the ability of a microservice to participate in more complex compositions should be considered from its design phase. This approach helps prevent future rework and ensures that the service can effectively collaborate with other components, contributing to the development of more robust, scalable, and integrated solutions.

2.1 Challenges in Microservice-Based Architectures

In microservice development, it is common to deal with the inherent complexity of managing multiple services alongside the heterogeneity of technologies and tools [8, 10]. Over time, microservice-based applications may exhibit issues related to quality, interoperability, and lack of cohesion among services implemented using different languages and technologies.

The use of API gateways or messaging systems can simplify communication and coordination between independent microservices; however, it introduces significant challenges in terms of configuration and implementation. Configuring an API (Application Programming Interface) gateway requires the precise definition of routing rules, security policies, and authentication mechanisms, which can become complex and error-prone, especially in dynamic environments where services are frequently added or modified [10]. Similarly, messaging systems require proper configuration of queues, topics, subscriptions, and message handlers, which can be difficult to manage in medium- and large-scale systems.

Ensuring compatibility across different API versions is also a challenge, requiring strict versioning and deprecation strategies [10]. Moreover, effective communication depends on well-defined and up-to-date documentation, enabling developers to correctly follow established standards. The absence of such documentation may lead to configuration errors and difficulties in integrating new services.

The use of containers enables the reuse and deployment of the same microservice across different environments, such as testing, staging, and production. However, each environment typically has specific requirements, including environment variables, database configurations, and network settings. Managing these variations for the same microservice across multiple environments can become a significant operational and deployment challenge.

Microservice architectures naturally promote technological heterogeneity, allowing the integration of services implemented in different programming languages and using diverse communication protocols, which can optimize performance and scalability [10]. However, this diversity also introduces risks that may compromise system cohesion and design consistency [8]. For instance, different teams may adopt conflicting conventions for error handling or communication patterns. Therefore, it is essential to establish common guidelines and practices to ensure consistency and interoperability across services, mitigating the risks associated with technological diversity and preserving the overall integrity of the system.

2.2 Related Work

Given the challenges discussed, there is a clear need for an intermediate representation capable of abstracting structural and operational aspects of microservices in a standardized manner, without compromising the implementation autonomy of each service. Such a representation should act as a structural contract, promoting consistency in how services are defined, organized, and exposed.

Furthermore, the absence of automated mechanisms for applying these standards tends to increase divergence between projects, particularly in heterogeneous environments. In this context, it becomes essential to adopt an approach that not only enables the unified description of such patterns but also supports their automation. Domain-Specific Languages (DSLs) have been identified as a promising alternative, as they allow the declarative and structured modeling of domain-specific elements [7].

Based on a systematic literature review, in which 830 studies were initially identified and 38 were selected for detailed analysis after applying selection criteria, it was observed that few approaches address standardization and automation in an integrated manner within microservice architectures. In particular, there is a gap in solutions that are simultaneously generic, extensible, and technology-agnostic, especially in environments involving multiple languages and frameworks.

Additionally, although existing works explore DSLs and automation, they are mostly focused on specific domains such as the Internet of Things [5], embedded systems, or infrastructure automation [12], and do not directly address the challenges of standardization and integration in microservice-based systems considered in this study.

Among the works that are closer to the proposed approach, two stand out: *Developing Microservice-Based Applications Using the Silvera Domain-Specific Language* [13] and *Blueprint: A Toolchain for Highly-Reconfigurable Microservices* [2].

Silvera [13] is a DSL designed for the declarative modeling of microservice-based architectures, enabling the specification of elements such as services, gateways, and endpoints, from which code can be automatically generated in different programming languages. In contrast, Blueprint [2] proposes a toolchain-based approach for configuring, building, and deploying microservices, leveraging abstractions and an intermediate representation to enable artifact generation and application reconfiguration.

While both approaches demonstrate the use of abstraction and automation to reduce complexity in distributed systems development, they primarily focus on initial code generation and structural configuration. They do not fully address the integrated automation of multiple configuration scenarios throughout the **software lifecycle**, nor do they emphasize generality in heterogeneous environments and the standardization of diverse configuration aspects.

In many practical development environments, standardization processes still depend heavily on manual supervision and on the expectation that all developers consistently follow architectural and organizational conventions previously defined by technical leaders or software architects. This often leads to recurrent verification activities, manual code reorganizations, and inconsistencies between teams and projects, especially in distributed environments. By contrast, the approach proposed in this work seeks to embed these conventions directly into the generation and configuration process, allowing such standards to be automatically applied and propagated across projects. Based on the professional experience of the first author, it is expected that this type of automation can produce short-term gains in consistency and development productivity, although the approach still requires further validation in real industrial environments.

3 RI-Language and TSE

This work proposes two main contributions: (i) the RI language, a Domain-Specific Language (DSL) designed for the declarative description of microservices; and (ii) an associated tool, named TSE, responsible for interpreting RI specifications and automating the generation of software artifacts by associating semantic actions with the grammar productions of the language.

The proposed language enables the structured and technology-independent description of microservice projects, including their properties and specifications. Complementarily, the TSE tool, developed using language processing techniques, leverages the RI lexer and parser to interpret specifications and automate the implementation of modules and configurations according to the defined patterns [3]. This automation is achieved through code generation, file creation, and controlled modification of specific code segments, reducing manual effort and promoting greater consistency across projects.

3.1 RI Language

The RI language was conceived as a Domain-Specific Language (DSL) aimed at the structured description of microservices and their associated artifacts. Its primary goal is to provide a declarative, standardized, and easily interpretable representation. In this way, the language supports not only tool-driven automation but also understanding and manipulation by users with different levels of technical expertise. Additionally, RI serves as a form of living

documentation, facilitating the understanding of microservice structures, their dependencies, and configurations throughout the software lifecycle. Listing 1 presents a simple project declared using the RI language.

■ **Listing 1** Simple project declared in RI language.

```
project "ms-customer" {
  use nestjs
  git "https://github.com/user/ms-customer.git"
  config "default"
  folders clean_architecture
  deployment {
    version "0.0.1"
    port 3000
    url "http://localhost"
  }
  dependencies []
  controllers [
    {
      name "CustomerController"
      path "/customer"
      endpoints [
        {
          type REST
          name "getOneCustomer"
          pattern GET
          path "/:id"
        }
        {
          type REST
          name "createCustomer"
          pattern POST
          path "/"
        }
      ]
    }
  ]
}
```

The central construct of the language is the **project** block, which represents a project. This block encapsulates all relevant information required to describe a service within the tool's context. A project may include general metadata, infrastructure configurations, dependency definitions, and specifications of exposed interfaces. It comprises declarations that describe fundamental characteristics such as the adopted framework (**use**), the remote repository (**git**), predefined configurations (**config**), and the directory organization pattern (**folders**). The **deployment** section groups execution-related information, such as version, port, and base URL.

The **dependencies** section allows explicit declaration of relationships with other projects and plugins. Through this structure, it is possible to define service coupling explicitly, indicating which resources should be consumed and which plugins should be applied. This approach supports the automation of tasks involving multiple projects and enhances dependency traceability.

Furthermore, the language enables the specification of the interfaces exposed by the microservice through the `controllers` section. In this structure, controllers, their respective paths, and available endpoints are defined. Each endpoint may specify its type (e.g., REST), name, HTTP method (such as GET or POST), and associated path, enabling the automated generation of interface-layer components.

In general, RI was designed to be expressive and extensible, covering both structural and behavioral aspects of microservices.

3.2 RI Language Grammar

The RI language is defined through a formal grammar that specifies its syntactic structure, enabling interpretation through lexical and syntactic analysis techniques. Listing 2 presents a simplified version of the language grammar.

■ Listing 2 RI Grammar.

```
project      : "project" STRING "{" project_statements "}"

project_statements
    : project_statements project_statement
    | project_statement

project_statement
    : use_stmt
    | git_stmt
    | config_stmt
    | folders_stmt
    | deployment_stmt
    | dependencies_stmt
    | controllers_stmt

use_stmt     : "use" IDENTIFIER
git_stmt     : "git" STRING
config_stmt  : "config" IDENTIFIER
folders_stmt : "folders" IDENTIFIER

deployment_stmt
    : "deployment" "{"
      "version" STRING
      "port" NUMBER
      "url" STRING
    "}"

dependencies_stmt
    : "dependencies" "[" dependency_list "]"

dependency_list
    : dependency_list dependency_def
    | dependency_def

dependency_def
    : "{" "plugin" STRING "project" STRING "}"

controllers_stmt
    : "controllers" "[" controller_list "]"
```

```

controller_list
    : controller_list controller_def
    | controller_def

controller_def
    : "{"
      "name" STRING
      "path" STRING
      endpoints_def
    "}"

endpoints_def
    : "endpoints" "[" endpoint_list "]"

endpoint_list
    : endpoint_list endpoint_def
    | endpoint_def

endpoint_def
    : "{"
      "type" IDENTIFIER
      "name" STRING
      "pattern" IDENTIFIER
      "path" STRING
    "}"

```

The presented grammar defines the syntactic structure of the RI language, establishing how the different elements of a project must be organized. The entry point is the **project** block, which encapsulates all other declarations.

The language interpretation process is carried out in two main stages: lexical analysis and syntactic analysis. During lexical analysis, the source code is converted into a sequence of tokens, such as keywords (**project**, **use**, **deployment**), identifiers, and literal values. Subsequently, syntactic analysis uses these tokens to construct a structured representation of the project, ensuring that the input conforms to the rules defined by the grammar [1].

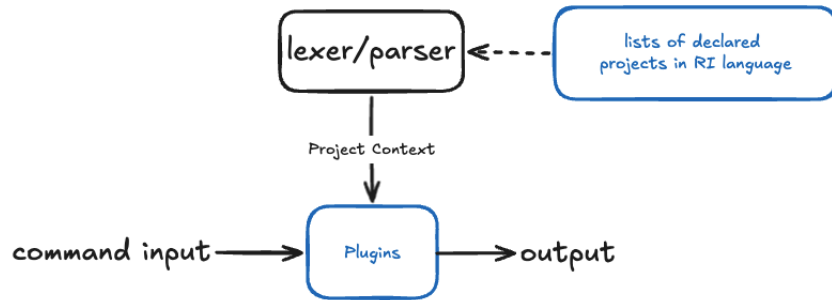
Based on this structured representation, the TSE tool is able to manipulate model elements, enabling the automated generation and modification of software artifacts. In this way, the formal definition of the grammar serves as the foundation for the standardization and extensibility of the RI language.

3.3 TSE (Toolbox Service Executor)

The Toolbox Service Executor (TSE) is the tool responsible for interpreting and executing the definitions specified in the RI language. Based directly on the language lexer and parser, TSE associates semantic actions with the recognized grammar structures, enabling declarative specifications to be transformed into concrete operations over software artifacts.

Operating through a command-line interface, TSE allows project instantiation from **.ri** files as well as the execution of plugins over these projects. When processing a file, the tool performs lexical and syntactic analysis, generating a structured representation that serves as context for subsequent executions.

Once instantiated, a project is stored in the tool's internal repository and can be used as a reference in future executions. When a plugin is invoked, TSE retrieves the corresponding project context, processes the RI definition, and provides the resulting structured information to the plugin, as illustrated in Figure 1.



■ **Figure 1** Simplified TSE workflow.

Plugins constitute the main extension mechanism of TSE and are defined through standardized contracts and interfaces. In general, each plugin receives a structured context derived from the RI specification, along with user-provided parameters, and produces a set of actions to be executed.

These actions are not applied directly by the plugin itself but are mediated by a standardized result structure that encapsulates operations such as file creation, content modification, and directory organization. This approach ensures greater control over execution, promoting predictability, consistency, and isolation between different plugins.

Interaction with the tool occurs through commands that specify the plugin to be executed and, optionally, the target project context. This model enables different plugins to operate over the same declarative base, supporting consistent and context-aware automation.

The TSE command-line interface is defined in Listing 3.

■ **Listing 3** TSE command-line interface.

```

Usage:
  tse [-l] [--help]
  tse <input_file.dsl>
  tse <plugin_name> [-p] [-o <output_path>] [-f <input_file>]
  tse <plugin_name> <project_name> [-p] [-o <output_path>]

Flags:
  -p          Preview of plugin execution
  -o          Output directory path
  -f          Target file path for plugin-specific use
  -l          List all available plugins
  --help      Display help information
  
```

4 Case Study

This section presents a simplified example of the proposed approach, considering the existence of a previously defined service and a set of available plugins in the platform.

4.1 Available Plugins in TSE

The TSE tool provides a set of extensible plugins designed to operate over projects defined in the RI language. In this case study, two main plugins are highlighted:

- **NestJsService**: responsible for generating HTTP communication modules based on NestJS from a declarative microservice definition. The plugin interprets controllers and endpoints defined in the context and automatically generates typed services for external consumption.
- **EnvDeployment**: responsible for extracting deployment information from a microservice and injecting it into `.env`-type configuration files, enabling standardization of environment variables across different consumer projects.

These plugins are maintained within the TSE ecosystem and operate over projects instantiated through RI definitions, making them available for automated processing and integration tasks.

4.2 Initial Context

Consider a scenario in which a microservice named **ms-users**, responsible for user management, already exists within the organization. This service was developed by a separate team using Python with FastAPI, and its structure was defined in a file named *ms-users.ri* using the RI language and subsequently instantiated through the TSE tool.

```
tse ms-users.ri
```

After being processed through the RI lexical and syntactic analysis pipeline, the resulting definition is stored in the TSE repository. It can later be extended with new declarations or used as input for plugin execution.

The RI file content is shown in Listing 4:

■ **Listing 4** User microservice project declared in RI.

```
project "ms-users" {
  use FastAPI
  git "https://github.com/user/ms-users.git"
  config default
  folders clean_architecture
  deployment {
    version "1.0.0"
    port 3001
    url "http://localhost"
  }
  dependencies []
  controllers [
    {
      name "UserController"
      path "/users"
      endpoints [
        {
          type REST
          name "getUserDataById"
          pattern GET
          path "/:id"
        }
      ]
    }
  ]
}
```

In parallel, a new team starts the development of a microservice named **ms-reports**, responsible for report generation. This service is implemented in TypeScript using NestJS and is also initially described using the RI language, as shown in Listing 5:

■ **Listing 5** Reports microservice project declared in RI.

```
project "ms-reports" {
  use NestJS
  git "https://github.com/user/ms-reports.git"
  config default
  folders clean_architecture
  deployment {
    version "0.0.1"
    port 3000
    url "http://localhost"
  }
}
```

During development, a requirement emerges for **ms-reports** to consume data from the **ms-users** microservice. In a traditional approach, this integration would require manual implementation of HTTP clients, definition of communication contracts, and configuration of environment variables. Additionally, differences in technology stacks – such as programming languages and frameworks – often lead to inconsistent integration strategies across teams.

4.3 Scenario for Automatic Integration and Configuration Generation

In the proposed approach, these challenges are mitigated by using the RI language as a declarative source capable of being interpreted by the tool. From the analysis of the defined structures, the tool is able to infer relevant information such as endpoints, routes, and deployment configurations, making them available to plugins that automate the generation of the necessary artifacts for service integration.

Following the example, the development team responsible for **ms-reports** executes the following command:

```
tse NestJsService ms-users
```

The tool interprets the declarative definition of the **ms-users** project and automatically generates the artifacts required for its integration, including communication modules, route definitions, and standardized access structures.

As a result of this process, the plugin creates an organized directory and file structure, as shown in Listing 6:

■ **Listing 6** Directory structure and files automatically generated.

```
user-api-service/
| dto/
| |_ index.ts
|- user-api.module.ts
|_ user-api.service.ts
```

As a result, the *UserApiServiceModule* is generated, encapsulating all HTTP communication with the **ms-users** microservice. The **dto** directory centralizes data transfer objects, while the **module** and **service** files represent, respectively, the module configuration and the service integration logic associated with the user service.

One of the main generated artifacts is the *user-api.service.ts*, which implements the actual calls to the service endpoints. This file is automatically generated from the RI definition, without requiring manual modifications. An example of its content is shown in Listing 7.

■ **Listing 7** Contents of the *user-api.service.ts* file.

```
import { HttpService } from "@nestjs/axios";
import { BadRequestException, Injectable } from "@nestjs/common";
import { ConfigService } from "@nestjs/config";
import { firstValueFrom } from "rxjs";

@Injectable()
export class UserApiService {
  private readonly baseUrl;
  private readonly url;

  constructor(
    private readonly httpService: HttpService,
    private readonly configService: ConfigService,
  ) {
    this.baseUrl = this.configService.get<string>("MS_USERS_BASE_URL");
    this.url = `${this.baseUrl}/users`;
  }

  async getUserById(id: string): Promise<any> {
    try {
      const response = await firstValueFrom(
        this.httpService.get(`${this.url}/${id}`),
      );
      return response.data;
    } catch (error) {
      throw new BadRequestException(error, "Error at getUserById");
    }
  }
}
```

In addition, the **EnvDeployment** plugin can be used to automatically inject service deployment configurations into the consumer project environment, further eliminating the need for manual configuration.

The **EnvDeployment** plugin can be executed using the following syntax:

```
tse EnvDeployment ms-users -f .env.local
```

In this case, the plugin automatically inserts the deployment information of the selected microservice into the environment file, ensuring standardization across consumer systems.

An example of the updated *.env.local* file is shown in Listing 8:

■ **Listing 8** Content of the *.env.local* file.

```
MS_REPORTS_DEPLOYMENT_VERSION=0.0.1
MS_REPORTS_DEPLOYMENT_PORT=3000
MS_REPORTS_DEPLOYMENT_URL=http://localhost

S3_ENDPOINT=*****
S3_ACCESS_KEY=*****
S3_SECRET_KEY=*****
S3_PUBLIC_URL=*****
S3_APP_BUCKET=*****
S3_REGION=*****

RESEND_API_KEY='*****'
```

```

SYSTEM_EMAIL_ADDRESS='***** <example@gmail.com>'

# BEGIN MS_USERS_DEPLOYMENT
MS_USERS_DEPLOYMENT_VERSION=1.0.0
MS_USERS_DEPLOYMENT_PORT=3001
MS_USERS_DEPLOYMENT_URL=http://localhost
# END MS_USERS_DEPLOYMENT

```

It can be observed that the file now contains a clearly delimited section automatically generated by the plugin, containing the deployment information of the **ms-users** microservice. If these values are updated in the original service definition, the plugin can be re-executed in an idempotent manner, updating only the marked section while preserving the rest of the file.

4.4 Case Study Conclusion

The presented scenario demonstrates how the use of the RI language, combined with plugin execution in TSE, significantly simplifies the integration process between software projects. In the described example, the team responsible for **ms-reports** was able to establish communication with **ms-users** without manually implementing HTTP clients, data contracts, or environment configurations.

From the declarative definition of the existing service, the tool was able to automatically infer the required information and generate both integration modules and the associated configuration artifacts. This process reduces development effort and helps avoid inconsistencies commonly found between teams, especially in environments involving different technologies and implementation standards.

Furthermore, the example illustrates how the approach promotes the reuse of definitions and consistency across services. If changes occur in the **ms-users** specification, the corresponding artifacts can be automatically updated, keeping consumer services aligned without requiring manual rework.

This case study demonstrates, in a practical scenario, how the combination of a declarative DSL with extension mechanisms based on semantic actions can support standardization, automation, configuration management, and integration in microservice-based architectures.

5 Conclusion and Future Work

This work presented the RI language and the TSE tool as an approach for declarative specification and automated artifact generation in microservice-based architectures. Through a set of heterogeneous microservices, it was demonstrated how the platform enables project instantiation, as well as the automatic generation of integration modules and operational configurations through extensible plugins.

As directions for future work, the first and most immediate step involves validating the proposed approach in real industrial environments, allowing the RI language and the TSE platform to be evaluated under realistic development and maintenance scenarios. Although the current evaluation was conducted as a proof of concept, the RI language has already been informally presented to members of the first author's development team, where the general idea was positively received and tested through small-scale experiments. These initial interactions indicate potential practical applicability, particularly regarding standardization and automation of repetitive configuration tasks. Nevertheless, a longer evaluation period in real projects is still necessary in order to assess usability aspects, identify limitations, and observe how the approach behaves during the continuous evolution of distributed systems.

In parallel, there is also a clear need to increase the expressiveness of the RI language, enabling more detailed modeling of architectural and operational aspects of microservices. These extensions include explicit definitions for databases, persistence strategies, Docker container configurations, network policies, and service-to-service communication, as well as richer controller specifications, including formal input/output contracts and strongly typed service interfaces.

Additionally, a relevant extension of this approach is the implementation of reverse-engineering plugins, capable of analyzing existing projects and automatically converting code artifacts into RI language definitions. In this scenario, previously implemented structures such as controllers, services, and modules could be interpreted and transformed into equivalent declarative representations, significantly reducing the effort required for migration and adoption in legacy systems.

This reverse-engineering mechanism would allow the RI language to be used not only as a starting point for system generation, but also as a unifying model for representing existing software architectures. In this sense, the TSE tool would evolve into a bidirectional transformation environment, supporting both the generation and extraction of architectural definitions. This capability significantly increases the potential for adoption, especially in industrial contexts where legacy systems are a common barrier to introducing new architectures and development standards.

Within the TSE ecosystem, there is also significant potential for expanding the plugin infrastructure, since its extensible architecture enables new automations without modifying the core system. This opens space for continuous community contributions, enabling plugins for infrastructure-as-code generation, cloud provider integration, observability, automated testing, and deployment orchestration.

Therefore, the combination of an extensible DSL and a plugin-based execution platform positions the proposed approach as an environment open to incremental evolution, both at the language level and at the level of software engineering automation.

References

- 1 Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley, 1986.
- 2 Vaastav Anand, Deepak Garg, Antoine Kaufmann, and Jonathan Mace. Blueprint: A toolchain for highly-reconfigurable microservices. *Association for Computing Machinery*, pages 482–497, October 2023. doi:10.1145/3600006.3613138.
- 3 David Beazley. Ply (python lex-yacc), 2025. [Online]. Available: <https://www.dabeaz.com/ply/ply.html>. [Accessed: Jan. 8, 2026].
- 4 Morgan Bruce and Paulo A. Pereira. *Microservices in Action*. Manning Publications Co., 2019.
- 5 Jiajun Chen, Jiyi Wu, Yuan Zuo, Luwei Fu, and Zhiwei Zhao. Mar-dsl: A domain-specific language for iot systems implementation. In *2024 International Conference on Ubiquitous Computing and Communications (IUCC)*, pages 56–62, 2024. doi:10.1109/IUCC65928.2024.00025.
- 6 Thomas Erl. *SOA: Principles of Service Design*. Prentice Hall, 2007.
- 7 Martin Fowler. *Domain-Specific Languages*. Pearson Education, 2010.
- 8 David Gonzalez. *Developing Microservices with Node.JS*. Packt Publishing Birmingham, UK, 2016.
- 9 Pooyan Jamshidi, Claus Pahl, Nabor Mendonça, James Lewis, and Stefan Tilkov. Microservices: The journey so far and challenges ahead. *IEEE Software*, 35:24–35, May 2018. doi:10.1109/MS.2018.2141039.

- 10 Sam Newman. *Building Microservices*. O'Reilly Media, Inc., 2015.
- 11 Mark Richards and Neal Ford. *Fundamentals of Software Architecture*. O'Reilly Media, 2020.
- 12 Wilson Valdez Solis, Priscila Cedillo, and Attila Kertesz. Micaal: A domain-specific language for microservices in ambient assisted living. *IEEE Access*, 13:56255–56272, 2025. doi: 10.1109/ACCESS.2025.3555831.
- 13 Alen Suljkanovic, Branko Milosavljevic, Vladimir Indic, and Igor Dejanovic. Developing microservice-based applications using the silvera domain-specific language. *Applied Sciences*, 12(13):6679, 2022.