

Beyond Standard English: L1-Aware Benchmarks for Evaluating API-Calling in LLM Agents

Vitor Gaboardi dos Santos ✉ 

School of Computing, Dublin City University, Ireland

Boualem Benatallah ✉ 

School of Computing, Dublin City University, Ireland

Rinn AI Research Ireland Centre on AI and Data Science, Dublin, Ireland

Silvana Togneri MacMahon ✉ 

School of Computing, Dublin City University, Ireland

Abstract

Background. Users can express the same request in many ways when interacting with LLM agents, particularly non-native English speakers whose writing reflects linguistic patterns transferred from their first language (L1). Existing benchmarks for evaluating API usage in LLM-based software systems typically rely on standardized English and overlook linguistic diversity. At the same time, collecting such data from users with diverse linguistic backgrounds is costly and difficult to scale.

Aims. This study proposes L1-AUG, an L1-aware utterance generation method, and investigates three research questions: whether LLMs can generate API-calling utterances reproducing linguistic patterns of non-native English speakers; whether L1-aware utterance generation leads to higher linguistic diversity; and whether API-calling utterances that simulate non-native English linguistic patterns are more challenging for LLM agents to solve.

Results. L1-AUG can generate utterances reflecting linguistic patterns extracted from essays written by non-native English speakers; L1-AUG generates benchmarks with higher linguistic diversity; and LLM agents achieve lower performance on the benchmark created using L1-AUG.

Conclusions. Conditioning benchmark generation on linguistic patterns extracted from non-native English essays produces benchmarks that better reflect users with diverse linguistic backgrounds. Our results indicate that LLM agents struggle more with such inputs, suggesting that training on linguistically diverse datasets may be necessary to improve robustness in API-calling tasks.

2012 ACM Subject Classification Software and its engineering → Software testing and debugging; Computing methodologies → Natural language processing

Keywords and phrases Software Testing Benchmarks, LLM-based Software Agents, Linguistic Diversity, L1 Transfer Patterns

Digital Object Identifier 10.4230/LIPIcs.ESEM.2026.23

Category Technical Track Paper

Supplementary Material *Software:* <https://github.com/vitorgaboardi/l1-apg>
archived at `swb:1:dir:d1a5802cedd669ca0a47630a0e5662cdb9fd34b1`

Funding This publication has emanated from research conducted with the financial support of Taighde Éireann-Research Ireland under Grant No. 18/CRT/6223.

1 Introduction

Large Language Model (LLM) agents have been widely adopted in conversational AI, serving as assistants that fulfill user utterances by connecting LLMs to external tools via APIs, reasoning over intermediate steps, and maintaining conversational memory [32]. APIs expand the scope of standalone LLMs by giving them access to up-to-date information (e.g., exchange rates, weather forecasts), control over physical environments (e.g., robot control), or the



© Vitor Gaboardi dos Santos, Boualem Benatallah, and Silvana Togneri MacMahon;
licensed under Creative Commons License CC-BY 4.0

20th International Symposium on Empirical Software Engineering and Measurement (ESEM 2026).

Editors: Robert Feldt, Maria Paasivaara, Daniel Mendez, Stefan Wagner, and Marvin Muñoz Barón;

Article No. 23; pp. 23:1–23:20



Leibniz International Proceedings in Informatics

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

ability to orchestrate cross-platform digital workflows [29]. As these systems increasingly interact with software services, ensuring LLM agent robustness becomes a major challenge. Nevertheless, LLM agents’ effectiveness depends on their ability to interpret different user intents and map them to structured API calls [19].

Given the richness of language, users can express the same request in many different ways, including variations in vocabulary, phrasing, sentence structure, and writing style [24, 20]. These variations are influenced by users’ linguistic backgrounds, as speakers may transfer linguistic conventions from their first language (L1) to their second language (L2) writing, a phenomenon known as L1 transfer [15, 1]. For instance, Mandarin speakers writing in English may omit articles and auxiliary verbs, since neither feature is grammaticalized in Mandarin. However, recent studies have shown that LLMs are not robust to lexical and syntactic variations [16] and perform worse when processing text generated by non-native English writers [18, 23]. This limitation is particularly concerning given the scale of L2 English use: English is the second language of over one billion speakers, compared to approximately 400 million native speakers [25]. Besides, widely used API-calling benchmarks, such as BFCL [19] and ToolBench [21], do not explicitly evaluate the effect of L1-aware linguistic diversity on API use. Ensuring robustness to diverse utterances is essential for reliable LLM agents and for quality assurance in software systems that integrate them, as untested linguistic diversity can cause service failures and degrade user experience.

Crowdsourcing is an effective method for obtaining diverse utterances, where humans are recruited via online platforms to curate benchmarks [22]. Including people from different backgrounds, countries, cultures, and domain knowledge enables the collection of diverse linguistic data that reflects real-world language variations [34]. Furthermore, each person carries distinct linguistic patterns, defined as systematic and recurring features of language use that collectively represent a “signature” of a person’s writing style, such as lexical and syntactic preferences, punctuation habits, and preferred sentence lengths [14, 17]. However, crowdsourcing has limitations, including cheating, coordination overhead, long collection times, and high labor costs, which make scaling benchmark construction difficult [33].

LLMs have been used to address crowdsourcing’s scalability limitations, enabling automatic generation of API-calling benchmarks. However, traditional LLM-based benchmark generation often lacks the linguistic diversity associated with text provided by real users and tends to produce common English utterances. Recent approaches [5, 7] attempt to increase diversity by conditioning LLMs on personas, defined as user profiles combining sociodemographic attributes (e.g., native language, age, gender) and psychosocial attributes (e.g., occupation, hobbies, interests) [28]. Although existing persona-based approaches improve diversity, they rely primarily on abstract attribute labels (e.g., “native language: Mandarin”, “occupation: engineer”) rather than actual linguistic behavior. We hypothesize that representing personas with only abstract attribute labels limits LLMs’ ability to capture linguistic patterns associated with L1 transfer in non-native English writing, because such representations are not grounded in text generated by real users.

In this study, we propose the L1-Aware Utterance Generation Method (L1-AUG), a data-driven approach for generating API-calling utterances that reflect linguistic patterns extracted from essays written by non-native English speakers. Unlike existing approaches that rely only on sociodemographic and psychosocial attributes, L1-AUG conditions utterance generation on sociodemographic attributes and linguistic patterns extracted from real non-native English writing. L1-AUG first uses a Chain-of-Thought (CoT) [30] prompting strategy to extract recurring linguistic patterns, such as common vocabulary choices and grammatical mistakes, from essays written by people from 20 countries from the ICNALE corpus [13]. Then, L1-AUG uses the extracted linguistic patterns to steer utterance and API-call generation.

The proposed approach enables the creation of datasets and benchmarks that support (1) evaluating the robustness of LLM agents to diverse utterances and (2) fine-tuning LLMs to improve their robustness to diverse utterances. We evaluate the performance of an LLM agent in fulfilling the generated utterances, allowing us to assess whether these utterances pose greater challenges for agents. We investigate the following research questions (RQs):

- **RQ1: Can LLMs reproduce linguistic patterns extracted from essays written by non-native English speakers when generating API-oriented utterances?** LLMs tend to generate text aligned with native English conventions, reflecting the distribution of their training data [9]. In this RQ, we investigate (1) whether utterances generated by L1-AUG reproduce linguistic patterns extracted from essays written by non-native English speakers, and (2) whether conditioning generation on personas associated with different countries results in distinct linguistic patterns. If successful, this would support using LLMs to generate diverse benchmarks that reflect non-native English writing while providing a scalable alternative to human data collection. We generate utterances and API calls for 10 APIs across different domains using 5 LLMs.
- **RQ2: Do utterances generated by L1-AUG provide higher linguistic diversity compared to other generation methods?** In this RQ, we investigate whether L1-AUG leads to higher linguistic diversity compared to two API-calling benchmark generation methods: (1) ToolAlpaca [27] (ALP), an approach that promotes diversity without persona conditioning; and (2) Persona ToolAlpaca (P-ALP), a persona-based approach conditioned on sociodemographic and psychosocial attributes obtained from PersonaHub [5]. We hypothesize that grounding utterance generation in real linguistic behavior extracted from learner essays leads to more linguistically diverse utterances.
- **RQ3: Are utterances generated by L1-AUG more challenging for a general-purpose LLM agent to fulfill compared to utterances from other benchmarks?** In this RQ, we evaluate whether a general-purpose LLM agent can fulfill utterances generated by the ALP, P-ALP, and L1-AUG methods. Our goal is to investigate whether utterances generated by L1-AUG are more challenging for an LLM agent to fulfill compared to utterances generated using other generation methods. To ensure the generalizability of our findings, we evaluate three LLM backbones: Ministral-3-8B, Llama-3.1-8B, and Qwen3.5-9B.

Our results show: (1) L1-AUG can generate utterances that reproduce linguistic patterns extracted from essays written by non-native English speakers from different countries; (2) L1-AUG generates benchmarks with higher linguistic diversity compared to P-ALP and ALP; and (3) L1-AUG generates utterances that are more challenging for LLM agents to fulfill than those generated by P-ALP and ALP.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 presents the proposed L1-AUG pipeline. Section 4 describes the experimental setup, results, and analysis for each RQ. Finally, Section 5 concludes the paper and outlines directions for future research.

2 Related Work

Patil et al. [19] introduced BFCL, a function-calling benchmark for evaluating LLM agents' ability to invoke APIs and generated using crowdsourcing and LLMs. Qin et al. [21] generated ToolBench, an LLM-generated API-calling dataset covering utterances that require invoking a single API or coordinating between multiple APIs. While these datasets are widely used to evaluate LLMs' API-calling capabilities, they rely on conventional English and do not explicitly account for linguistic diversity.

Other works introduced mechanisms to improve diversity. Tang et al. [27] generated utterances using guidelines to encourage the use of different syntactic patterns and avoid verb repetition. Santos et al. [24] used the *logit bias* parameter of LLMs to encourage lexical diversity by penalizing frequently used tokens when paraphrasing utterances. Berro et al. [2] showed that GPT-3.5 can generate syntactically diverse paraphrases by simulating crowdsourcing workflows used for collecting utterances in task-oriented dialogue systems. While these approaches improve lexical or syntactic diversity, they do not incorporate personas during utterance generation, limiting the generation of utterances that reflect rich user contexts, such as occupation, hobbies, cultural background, or writing style.

Other works generated API-oriented utterances by simulating personas with different characteristics. Huang et al. [12] represented personas with different emotional states (e.g., happiness, anger, or sadness); Xu et al. [31] used mood, gender, English dialect, and age to simulate personas; and Cheng et al. [7] incorporated demographics (e.g., gender, age, weight, profession), personal interests (e.g., enjoying exercise), preferences (e.g., user prefers to buy at Amazon), and environmental context (e.g., city, date, time, weather). While these methods generate utterances that reflect richer contexts and improve linguistic diversity, they rely on abstract attribute labels to represent personas and do not incorporate examples of text generated by real users. Consequently, they do not explicitly capture how users from different linguistic backgrounds write when interacting with conversational systems.

Gao et al. [8] showed that LLMs can generate English dialogues with linguistic patterns influenced by a speaker’s L1 when conditioned on real examples. However, their work focuses on multi-turn spoken dialogue generation between native and non-native English speakers beyond the API-calling domain. Also, they focus on spoken language, including hesitation markers such as “um” or “uh”. In contrast, our work focuses on the API-calling domain and generates single-turn utterances grounded in written language.

Prior work on generating API-calling benchmarks has focused on improving lexical and syntactic diversity or on conditioning generation on personas represented by abstract attribute labels, such as age, gender, and occupation. However, these approaches are not grounded in text written by real people and do not capture how non-native English speakers write. Consequently, the resulting benchmarks do not evaluate how such linguistic variation may affect LLM agent performance. To address this gap, we propose a pipeline that conditions LLM generation on linguistic patterns extracted from essays written by English learners from 20 different countries. Although these patterns are extracted from argumentative essays rather than conversational interactions, they provide a data-driven alternative to abstract persona attributes for conditioning utterance generation.

3 Generation of API-oriented utterances and API calls

In this section, we first describe the preprocessing of the ICNALE corpus [13], which we use as seed data for generating personas. We selected ICNALE because it provides sociodemographic attributes (e.g., age, gender, occupation, and country) and also essays written by real users. Next, we describe our proposed L1-AUG method.

3.1 ICNALE Corpus Processing and Analysis

The ICNALE [13] includes a corpus of English essays written by English learners from 28 countries. Each writer produced two essays on the following topics: (1) “part-time jobs for college students” and (2) “non-smoking at restaurants.” The dataset includes metadata such as gender, age, academic major/occupation, and proficiency level. Essays are available as .txt files with filenames encoding writer metadata, including country, topic, unique ID, and proficiency.

We pre-processed ICNALE and produced a CSV file with the following data for each writer: country, L1, gender, age, major/occupation, proficiency level, and essays. We excluded writers from the United States, Great Britain, and Australia to focus on non-native English speakers. We also excluded countries with fewer than 40 essays to ensure enough seed essays for utterance generation. The final dataset includes 20 countries with the following sample sizes: Bangladesh (101 people), Brunei (50), Cambodia (52), China (400), Hong Kong (100), India (191), Indonesia (200), Japan (400), South Korea (300), Laos (40), Malaysia (107), Myanmar (132), Nepal (50), Pakistan (200), Philippines (200), Singapore (200), Taiwan (200), Thailand (400), United Arab Emirates (100), and Vietnam (438).

We analyzed the preprocessed ICNALE corpus to determine whether essays written by English learners from different countries show distinct linguistic patterns. We treat speakers from different countries as representing distinct linguistic backgrounds, even when they share the same native language. This design choice reflects the fact that linguistic diversity in non-native English writing may also be influenced by educational, cultural, and sociolinguistic factors of a given country beyond the speaker's L1.

We represent essays written by the same person as a feature vector composed of lexical and syntactic features commonly used in the literature [11, 25]. More precisely, we used the following features: (1) average words per sentence; (2) average characters per word; (3) root type-token ratio; (4) function word ratio, computed as the normalized frequency of function-word tokens based on the ONIX stop word lexicon (English)¹; and (5) 17 part-of-speech (POS) tag ratios: adjectives, ad positions, adverbs, auxiliary verbs, coordinating conjunctions, determiner, interjections, nouns, numerals, particles, pronouns, proper nouns, punctuation, subordinating conjunctions, symbols, verbs, and other.

Feature vectors were normalized using z -score normalization to ensure comparability across features with different numerical scales. We then computed pairwise cosine distances between all normalized vectors. Formally, for two essays i and j , with normalized vectors $\mathbf{z}_i$ and $\mathbf{z}_j$, the cosine distance is denoted by d_{ij} , where lower values indicate higher linguistic pattern similarity between essays. Now, let ℓ_i denote the country label associated with essay i . We define the set of within-country pairs as $D_{\text{within}} = \{(i, j) \mid i < j, \ell_i = \ell_j\}$ and the set of between-country pairs as $D_{\text{between}} = \{(i, j) \mid i < j, \ell_i \neq \ell_j\}$. The mean within-country distance is then defined as:

$$\bar{d}_{\text{within}} = \frac{1}{|D_{\text{within}}|} \sum_{(i,j) \in D_{\text{within}}} d_{ij}, \quad (1)$$

and the mean between-country distance is defined as

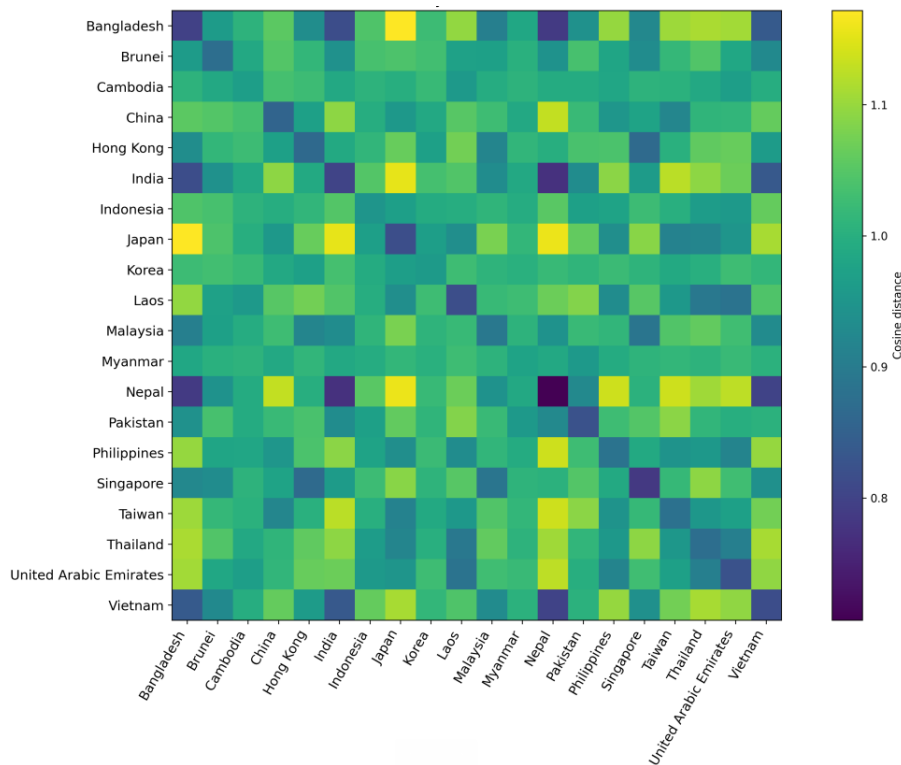
$$\bar{d}_{\text{between}} = \frac{1}{|D_{\text{between}}|} \sum_{(i,j) \in D_{\text{between}}} d_{ij}. \quad (2)$$

We test the hypothesis that essays from the same country show higher linguistic pattern similarities than those from different countries (i.e., $\bar{d}_{\text{within}} < \bar{d}_{\text{between}}$). Our results confirm this, with $\bar{d}_{\text{within}} = 0.861$ and $\bar{d}_{\text{between}} = 1.009$. To assess statistical significance, we ran a permutation test (1,000 iterations) in which we randomly shuffled country labels across essays while keeping pairwise distances fixed. Using the observed separation $\Delta_{\text{obs}} = \bar{d}_{\text{between}} - \bar{d}_{\text{within}} = 0.148$, we obtained $p < 0.01$, confirming statistical significance.

Figure 1 presents the average cosine distances for each country pair. Results show lower average cosine distance values along the diagonal, indicating that essays written by speakers from the same country are more similar to each other than those written by speakers from

¹ <https://eecs.csuohio.edu/~sschung/DSA460/StopwordList1.pdf>

different countries. Some South Asian countries (India, Nepal, and Bangladesh) show similar distances, consistent with their geographic proximity and shared exposure to British English. Japan shows higher distances overall, suggesting more distinctive L1 transfer patterns. Surprisingly, Thailand, the United Arab Emirates, and Laos also show low distances among themselves. These findings show that English learners from different countries show distinct linguistic patterns, motivating the design of L1-AUG.



■ **Figure 1** Average cosine distance between essays of different countries.

3.2 L1-Aware Utterance Generation Method

In this section, we present L1-AUG, our proposed method for generating utterances and API calls conditioned on linguistic patterns extracted from essays written by non-native English speakers. We first define a persona using sociodemographic and linguistic attributes derived from ICNALE [13]. Next, we use a CoT prompting method to analyze linguistic patterns in the provided essays and generate utterances conditioned on the extracted patterns and persona attributes. We adopt a CoT prompting strategy because it allows the model to explicitly identify recurring linguistic patterns in the essays before generating utterances. This intermediate reasoning step encourages generation conditioned on the extracted patterns rather than defaulting to grammatically standard English. We provide implementation details of L1-AUG in the remainder of this section.

3.2.1 Persona Definition

We define a persona P using the following template: “You are a [age]-year-old [gender] from [country] who speaks [language] with an English proficiency level of [level]. Your job involves working in [major]. These are examples of essays showing how you write: [reference_essays]”.

The placeholders are filled using data from ICNALE [13], although alternative data sources can also be used. This representation enables an LLM to condition its generation on the demographic and linguistic contexts associated with people from different countries.

3.2.2 Linguistic Patterns Extraction

Given a set of essays $E = \{e_1, e_2\}$ written by a persona P , an LLM is prompted to analyze and extract linguistic patterns of E before generating utterances and API calls. Specifically, we instruct LLMs to identify and output the following linguistic patterns from essays: (1) vocabulary choices that are uncommon in native English; (2) recurring syntactic structures; (3) spelling and grammatical errors; (4) false cognates; and (5) literal translations arising from L1-to-L2 transfer. These patterns were selected based on prior work in second language acquisition that maps and annotates texts written by non-native English speakers in terms of orthographic, lexical, morphological, grammatical, and L1 interference [1]. This step produces a set of descriptive linguistic patterns L associated with a persona’s writing style.

3.2.3 Utterance and API Call Generation

We designed instructions to guide the LLM in generating utterances that reflect the persona’s background, resemble real interactions with LLM agents, and reproduce linguistic patterns extracted from essays written by non-native English speakers.

We provide an LLM with a persona template and API documentation. Then, we instruct the LLM to generate utterances that reflect both the persona’s linguistic patterns and background while promoting diversity. Specifically, the LLM is instructed to role-play the provided persona, generate n utterances related to the provided API that reflect the extracted linguistic patterns (as explained in Section 3.2.2), incorporate parameter values (e.g., locations, currencies, and food items) based on the persona’s nationality, use the persona’s professional occupation as context when appropriate, and change lexical and syntactic choices across utterances while remaining consistent with the persona’s linguistic patterns.

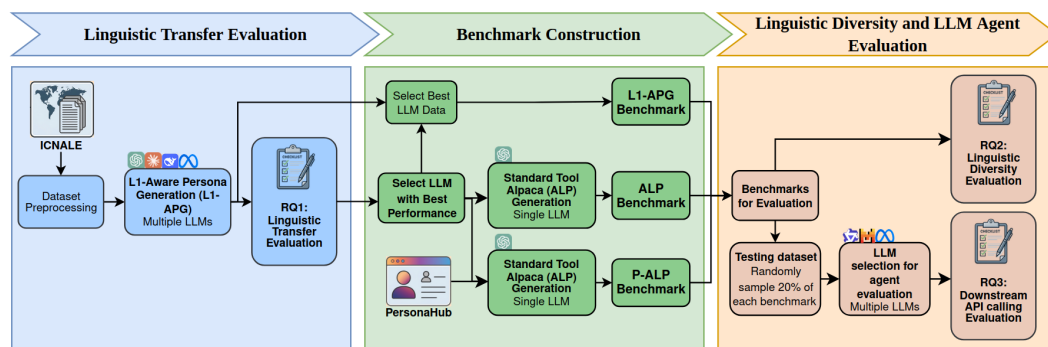
During an iterative prompt-development phase, we observed that some utterances reproduced conclusion markers commonly associated with argumentative essays, such as “In conclusion”, “To sum up”, and “As I mentioned”. We also observed API terminology that users do not commonly use when making requests. Therefore, we also instructed the LLM to generate utterances that resemble natural interactions with conversational agents, not incorporate essay-specific conclusion markers, to avoid technical API terminology (e.g., including “page number” or “API key” data), and use natural expressions such as “next Monday” rather than fixed formats like “2024-06-18”.

Finally, we prompted the LLM to generate the API calls needed to obtain the information required to answer each user utterance. To ensure API calls are valid, we included guidelines requiring arguments to follow the formats specified in the API documentation (e.g., ISO dates, standardized codes, and enumerated values) and forbidding the model from generating parameter values that cannot be inferred from utterances.

4 Experiments

Figure 2 provides an overview of our experimental design, which consists of three phases. First, we use L1-AUG with multiple LLMs to generate utterances and API calls that simulate linguistic patterns extracted from essays written by non-native English speakers across 20 countries. RQ1 evaluates whether the generated utterances indeed reproduce linguistic

patterns of the provided essays written by people from different countries. Second, we use the best-performing LLM in RQ1 to generate API-calling benchmarks using three methods: a no-persona baseline (ALP), a persona-based baseline (P-ALP), and the proposed L1-AUG method. RQ2 compares the linguistic diversity across these three benchmarks. Third, we evaluate a general-purpose LLM agent using 20% random samples of each benchmark. RQ3 compares the agent’s performance when deployed using three different LLMs, evaluating whether the benchmark generated by L1-AUG is more challenging than other benchmarks.



■ **Figure 2** Overview of experiment design.

4.1 RQ1: L1 Transfer Evaluation

In Section 3, we showed that essays written by non-native English learners from different countries show distinct linguistic patterns. In this section, we investigate whether L1-AUG can reproduce these linguistic patterns when generating API-oriented utterances.

4.1.1 Experimental Setup

We generate benchmarks using the L1-AUG method across 5 LLMs, 10 APIs, and 600 personas. In this subsection, we describe the data generation and evaluation processes.

LLMs. We generate utterances and API calls using five LLMs: GPT-5.4, Claude-4-Sonnet, DeepSeek-V4-Pro, GPT-4o-mini, and Llama-4-Maverick. We selected these models to cover both closed- and open-source LLMs with different inference costs and parameter scales. We use the OpenAI API² for generating data with GPT-5.4 and GPT-4o-mini, and the DeepInfra API³ for Claude-4-Sonnet, DeepSeek-V4-Pro, and Llama-4-Maverick. We use a temperature of 1.0 for all models.

APIs. We generate utterances and API calls for 10 APIs across different services: paper search (ArXiv), currency conversion, exercise retrieval, movie search, food nutrition lookup, tourist place search, restaurant search, stock price retrieval, current weather, and weather forecast. We selected APIs from different domains and with different numbers of input parameters to evaluate agents across tasks of different complexity while keeping the benchmark manageable for comparing generation methods and LLMs. While these APIs rely on existing

² <https://openai.com/api/>

³ <https://deepinfra.com/>

third-party services, we manually created documentation and developed Python wrapper functions for each API to handle request formatting, error checking, and data parsing. This interface allows an LLM agent to invoke APIs using function calls.

Data Generation. For each API, we randomly sample 3 personas from each of the 20 countries listed in Section 3.1, yielding 60 personas per API. Then, for each persona, we request 5 utterances and corresponding API calls. We use each persona only once per API generation process. We keep the persona set fixed across all LLMs to ensure fair comparisons. This process generates 300 utterances per API ($60 \text{ personas} \times 5 \text{ utterances} = 300 \text{ utterances}$), scaling to 3,000 utterances for each evaluated LLM across all 10 APIs. It also yields 150 utterances for each country ($3 \text{ personas} \times 5 \text{ utterances} \times 10 \text{ APIs} = 150 \text{ utterances}$). This distribution balances the dataset across countries while supporting meaningful analysis of non-native linguistic patterns.

Evaluation Metrics. We follow the same evaluation procedure described in Section 3. However, instead of computing feature vectors from the reference essays, we concatenate the five utterances generated for each persona and compute the same feature vectors using the lexical and syntactic features described in Section 3. Feature vectors are normalized using z -score normalization and compared using cosine distance. We report the same metrics defined in Section 3: $\bar{d}_{\text{within}}$, $\bar{d}_{\text{between}}$, $\Delta = \bar{d}_{\text{between}} - \bar{d}_{\text{within}}$, and a p -value from a permutation test (10,000 iterations) in which country labels are shuffled while keeping pairwise distances fixed. If $\bar{d}_{\text{within}} < \bar{d}_{\text{between}}$, utterances generated based on essays written by people from the same country are more similar to each other compared with utterances generated from essays written by people from different countries. This provides evidence that L1-AUG reproduces country-associated linguistic patterns extracted from the source essays.

We also conducted an exploratory qualitative analysis by sampling a subset of 500 generated utterances (100 per model). An annotator then manually evaluated these utterances to verify whether they simulate linguistic patterns consistent with the source reference essays. The annotator observed patterns in the essays, such as common misspellings and grammatical mistakes, typical sentence length, repeated words throughout the text, and unnatural phrasing, and evaluated whether these patterns were reproduced in the generated utterances. This analysis is a qualitative check, providing insight into whether the generated utterances show these linguistic patterns. We report the **Pattern Preservation Rate (PPR)**, defined as the proportion of utterances that preserve the linguistic patterns observed in the reference essays.

4.1.2 Results

We found that **L1-AUG generates utterances whose linguistic patterns are more similar to each other when conditioned on essays written by English learners from the same country than from those of different countries. Furthermore, qualitative analysis shows that GPT-5.4 had the best performance in reproducing linguistic patterns extracted from the reference essays, achieving the highest Δ (0.1165) and PPR (0.72).** Table 1 presents the results.

Results show different performance across models. GPT-5.4 and DeepSeek-V4-Pro achieved the two highest Δ values (0.1165 and 0.0893, respectively), indicating stronger country-level separation in generated utterances. Llama-4-Maverick and Claude-4-Sonnet produced moderate separation ($\Delta = 0.0658$ and 0.0570), while GPT-4o-mini achieved the lowest Δ (0.0144). Nevertheless, all models yielded $p < 0.001$, confirming that the observed

■ **Table 1** Cross-country linguistic pattern transfer evaluation across different LLMs.

LLM	$d_{\text{within}} \downarrow$	$d_{\text{between}} \uparrow$	$\Delta \uparrow$	$p\text{-value} \downarrow$	PPR $\uparrow$
GPT-5.4	0.8901	1.0060	0.1165	$p < 0.001$	0.72
DeepSeek-V4-Pro	0.9153	1.0045	0.0893	$p < 0.001$	0.26
Llama-4-Maverick	0.9373	1.0031	0.0658	$p < 0.001$	0.12
Claude-4-Sonnet	0.9460	1.0030	0.0570	$p < 0.001$	0.51
GPT-4o-mini	0.9871	1.0015	0.0144	$p < 0.001$	0.04

separation is statistically significant. In the qualitative analysis, GPT-5.4 achieved the highest PPR (0.72), while GPT-4o-mini achieved the lowest (0.04), consistent with the quantitative results. Claude-4-Sonnet achieved a relatively low Δ (0.0570) but a moderate PPR (0.51), suggesting that this model successfully generated utterances reproducing linguistic patterns extracted from the reference essays, but with lower country-level separation compared with the other models. DeepSeek-V4-Pro showed the opposite pattern, achieving a higher Δ (0.0893) but a lower PPR (0.26).

Table 2 shows examples of generated utterances. Example 1 includes many patterns that are present in the original essays, such as run-on sentences connected by commas, frequent use of the word “because”, plural and tense errors, and unusual phrasing (e.g., “split my concentration”). Example 2 includes over-intensification (“more than enough important”), unusual phrasing (“put a finger on”), and a preposition error (“fit for my routine”) that mirror patterns from the original essays. Example 3 shows preposition mistakes (“from Kyoto” instead of “in Kyoto”), a missing verb (“I interested”) and a plural error (“temple” instead of “temples”). Although all examples include linguistic patterns from the original essays, Examples 1 and 2 more closely reflect the structural and lexical patterns specific to the persona. In contrast, Example 3 contains isolated grammatical mistakes that are less distinctive of the original writing style.

■ **Table 2** Examples of generated utterances per model using L1-AUG.

#	Model	Utterance
1	GPT-5.4	Some of my friend in Jakarta talking about Pengabdi Setan, so I need that movie search, maybe only 5 result is enough because too many movies sometimes make me confused and split my concentration.
2	Claude-4-Sonnet	From my point of view, having strong core muscles is more than enough important for maintaining good posture during long working hours. I need to put a finger on some good abdominal exercises that fit for my routine using resistance bands.
3	GPT-4o-mini	Can you show tourist places from Kyoto? I interested in temple and some historical places there.

Finally, Figure 3 shows the average pairwise cosine distances between countries for utterances generated by GPT-5.4. The diagonal values are lower, indicating that utterances conditioned on linguistic patterns extracted from essays written by people from the same country are more similar to each other than to those from different countries. Furthermore, some interesting patterns emerge. Brunei, the Philippines, and Singapore show lower pairwise distances, likely reflecting shared patterns of Southeast Asian English resulting from their

shared history of English as an official language. Similarly, Nepal, Pakistan, and Bangladesh form a closer cluster, consistent with shared South Asian English features rooted in Indo-Aryan linguistic patterns. Japan and Korea also show low pairwise distances.

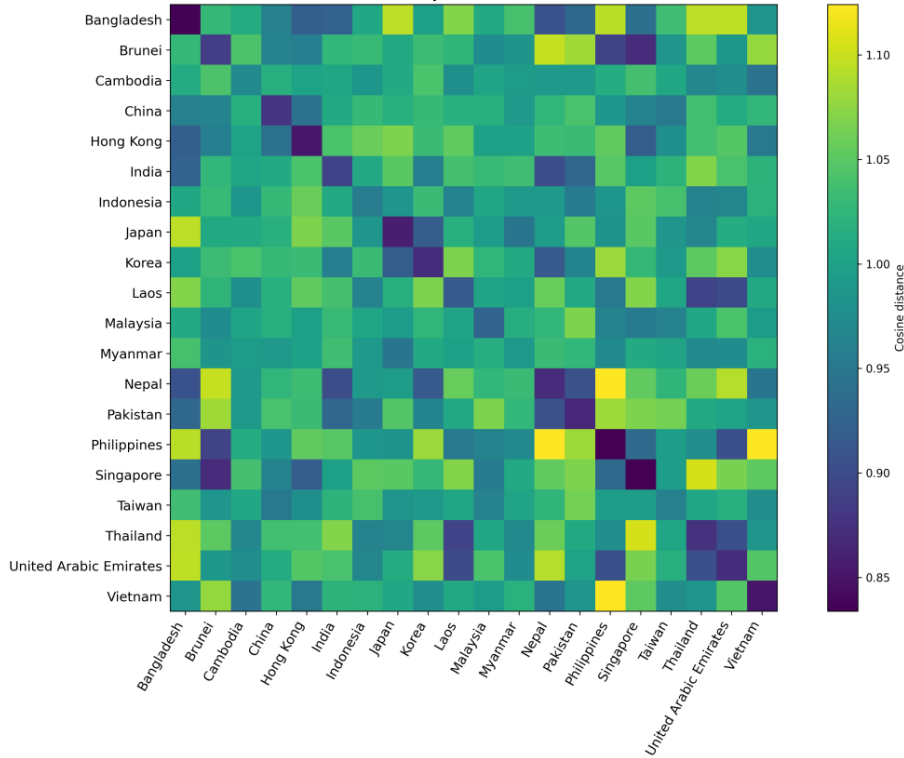


Figure 3 Average cosine distance between utterances generated by GPT-5.4 across personas from different countries.

4.2 RQ2: Linguistic Diversity Evaluation

In this section, we compare the linguistic diversity of benchmarks generated by three generation methods: (1) ToolAlpaca [27] (ALP), an approach that promotes diversity without persona conditioning; (2) Persona ToolAlpaca (P-ALP), a persona-based approach conditioned on sociodemographic and psychosocial attributes obtained from PersonaHub [5]; and (3) L1-AUG. Benchmarks with higher linguistic diversity reflect a broader range of ways in which users may express utterances, making them better suited for evaluating models.

4.2.1 Experimental Setup

We selected GPT-5.4 for benchmark generation because it achieved the highest performance in reproducing linguistic patterns extracted from essays written by non-native English speakers, as shown in Section 4.1.2. We generated utterances for the 10 APIs described in Section 4.1.1.

Generation Methods. To ensure a fair comparison across generation methods, we generated 300 utterances per API. For ALP and P-ALP, we achieve this by executing 60 programmatic calls to GPT-5.4 per API, with each prompt requesting a batch of 5 utterances (60 calls $\times$ 5 utterances = 300 utterances). The three evaluated generation methods are described as follows:

- **ALP**: This method uses the prompt proposed by ToolAlpaca [27], which instructs the LLM to generate utterances for a given API while promoting lexical and syntactic diversity. It does not include any persona conditioning.
- **P-ALP**: This method extends the ToolAlpaca [27] prompt by conditioning generation on persona descriptions sampled from PersonaHub [5]. Personas include attributes such as occupation, expertise, beliefs, and demographic characteristics (e.g., *a political analyst specialized in El Salvador’s political landscape*). For each generation iteration, we randomly sample one persona and instruct the LLM to incorporate the persona profile when generating utterances. Personas are not repeated across generations.
- **L1-AUG**: We use the benchmark generated by GPT-5.4 described in Section 4.1.1.

Evaluation Metrics. We use multiple established metrics based on prior work [10, 6, 36]. **Unique- n** [10] is the ratio of unique n -grams to total n -grams. We report Unique- n across unigrams, bigrams, and trigrams, allowing for evaluation of the proportion of new words and their combination across generation methods. **Syntax Similarity Mean (SSM)** [6] computes the average tree edit distance (TED) between constituency parse trees, computed over all pairwise utterance combinations per API using the top three parse tree levels. **Total Number of Words (TNW)** reflects the length of generated utterances. **Self-BLEU (SB)** [36] measures n -gram overlap within all utterances generated for the same API, where lower values indicate higher diversity. **Pairwise Semantic Similarity (PSS)** [10] computes one minus the average pairwise cosine similarity between utterance embeddings for the same API, as defined in Equation 3. The rationale is that utterances associated with the same API method should preserve the same intent while varying in writing style across personas with distinct backgrounds. Thus, lower pairwise cosine similarity indicates greater semantic diversity and results in higher PSS values. We selected these metrics because they provide complementary measures of lexical, syntactic, and semantic diversity.

$$\text{PSS} = 1 - \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \cos(\mathbf{e}_i, \mathbf{e}_j) \quad (3)$$

where n is the number of utterances for a given API and $\cos(\mathbf{e}_i, \mathbf{e}_j)$ is the cosine similarity between embeddings $\mathbf{e}_i$ and $\mathbf{e}_j$ of utterances $\mathbf{u}_i$ and $\mathbf{u}_j$, respectively. Embeddings are computed using Qwen/Qwen3-Embedding-4B [35], which was chosen given its high performance at the Massive Text Embedding Benchmark (MTEB) at the time of the experiments. We compute all metrics for each API and average them across APIs to obtain a single score.

4.2.2 Results

We found that **L1-AUG produced benchmarks with higher syntactic and semantic diversity, and a higher ratio of unique trigrams, while P-ALP achieved higher ratios of unique unigrams and bigrams. ALP showed the lowest diversity across all metrics.** These results support the hypothesis that conditioning utterance generation on L1 transfer patterns extracted from essays promotes linguistic diversity beyond what persona-augmented generation alone achieves. Table 3 presents the results.

P-ALP achieved the highest Unique-1 and Unique-2 scores (0.1761 and 0.4899, respectively), suggesting that using personas from PersonaHub [5] increases the diversity of unigrams and bigrams. We believe this is driven by the specificity of persona definitions, which prompt the model to use more context-specific vocabulary. For instance, a persona defined as “*A farmer who relies on disease-resistant crops for a successful harvest*” led to an utterance

■ **Table 3** Diversity evaluation of generated utterances for GPT-5.4.

Pipeline	TNW $\uparrow$	UNIQ1 $\uparrow$	UNIQ2 $\uparrow$	UNIQ3 $\uparrow$	SSM $\uparrow$	SB $\downarrow$	PSS $\uparrow$
ALP	4919.7	0.0504	0.1719	0.2924	5.5304	0.8497	0.3454
P-ALP	8169.6	0.1761	0.4899	0.6778	7.1950	0.4820	0.4337
L1-AUG	12879.0	0.0990	0.4785	0.7805	10.5188	0.3780	0.4611

containing “*disease-resistant tomato*”, representing an unusual wording that contributes to higher unigram diversity. Nevertheless, L1-AUG led on Unique-3 (0.7805), indicating that while P-ALP diversifies at the word level, L1-AUG produces more diverse trigram combinations. The baseline ALP scored the lowest across all Unique- n metrics.

L1-AUG achieved the highest SSM score (10.51), indicating higher diversity in sentence structure. P-ALP also improved over ALP (7.195 vs. 5.5304), suggesting that persona conditioning alone contributes to syntactic diversity. The large gap between ALP and L1-AUG highlights the impact of conditioning generation based on linguistic patterns extracted from real essays written by English learners from different countries.

L1-AUG led with the highest PSS (0.4611) and the lowest Self-BLEU (0.3780), indicating that its utterances include more diverse meanings with less n -gram overlap. ALP showed the highest Self-BLEU score (0.8497), suggesting that without persona or L1 conditioning, the model tends to generate semantically redundant utterances. P-ALP falls between the two methods, confirming that persona conditioning enhances semantic diversity but does not reach the level of diversity achieved by L1-AUG.

It is worth noting that L1-AUG produces longer utterances on average (TNW = 12879) compared to P-ALP and ALP. Increased utterance length expands the collection of n -gram combinations, which may contribute to higher Unique-3 scores. At the same time, longer utterances can increase word repetition and reduce the proportion of unique unigrams and bigrams, which may explain why P-ALP achieves higher Unique-1 and Unique-2 scores, and L1-AUG achieves the highest Unique-3 score. Although L1-AUG generates longer utterances, it achieves lower SB scores than ALP and P-ALP. Since longer utterances increase opportunities for n -gram overlap, this suggests that the diversity introduced by L1-AUG is not only a consequence of verbosity, but also reflects higher diversity across generated utterances. The increased utterance length is also consistent with findings from second-language research, which report that non-native speakers often use longer utterances and more explicit expressions when they lack shorter native-like phrases [3, 4]. These observations support the idea that L1-AUG achieves greater linguistic diversity than ALP and P-ALP.

4.3 RQ3: Evaluating the API-Use Capabilities of LLM Agents

In this section, we compare the performance of an LLM agent in fulfilling utterances generated by L1-AUG, P-ALP, and ALP. As shown in Sections 4.1 and 4.2, L1-AUG generates utterances that reproduce linguistic patterns extracted from essays written by non-native English speakers and achieves higher linguistic diversity. This experiment investigates whether these characteristics make the resulting utterances more challenging for LLM agents to fulfill.

4.3.1 Experimental Setup

We implement an LLM agent architecture and evaluate its deployment using three LLMs serving as the underlying reasoning engines. By testing with multiple LLMs, we ensure that our evaluation reflects the difficulty of the generated benchmarks rather than the limitations of any single LLM. The remainder of this section describes the agent implementation, LLMs, the testing dataset, and evaluation metrics.

LLM Agent Architecture. The agent is designed to process user utterances that require external APIs. The LLM agent operates as a general-purpose assistant with access to the ten APIs detailed in Section 4.1.1. The agent operates in an iterative loop. At each step, the agent can execute one or more API calls to gather the necessary information to answer the user’s utterance or return a final answer. API responses are appended to the agent’s conversation history, allowing it to reason over intermediate results and dynamically plan its next actions. We set a maximum execution of five iterations to prevent the agent from looping indefinitely while allowing recovery from intermediate errors.

LLM Selection and Deployment. We evaluated three LLMs as the agents’ reasoning models: Ministral-3-8B, Llama-3.1-8B, and Qwen-3.5-9B. We chose these LLMs because they support native tool-calling and are open-source, allowing local deployment. To ensure a fair comparison and accommodate local computational constraints, we restricted our selection to models with fewer than 10 billion parameters. We deployed all models locally on a machine with a single NVIDIA GeForce RTX 4090 GPU (24 GB VRAM) and an Intel Core i9 CPU (24 cores, up to 5.8 GHz).

Testing Dataset. We randomly sampled 20% of the data from each benchmark generated by ALP, P-ALP, and L1-AUG to build the test set. This selection yields 600 utterances and their corresponding ground-truth API calls per benchmark. This sampling strategy provides broad coverage while keeping evaluation costs manageable. Because our agent executes generated API calls against real API backends during evaluation, running experiments on the full benchmark would require at least 27,000 API calls ($3,000 \text{ utterances} \times 3 \text{ benchmarks} \times 3 \text{ LLMs}$), making evaluation computationally expensive and time-consuming. To ensure balanced representation, we apply stratified sampling [26] across API for all benchmarks. For L1-AUG, we also stratify by country to ensure equal representation across linguistic backgrounds. This configuration results in 60 test instances per API for each benchmark.

Evaluation Metrics. We evaluate the LLM agent’s API-calling capabilities by verifying whether its generated API calls execute successfully and match the ground-truth API calls. Accurate API calls ensure that the agent retrieves the correct data required to fulfill the user’s utterance. We measure performance using the following metrics: (1) **Execution Success Rate (ESR)**: Measures the proportion of total API calls generated by the agent that execute successfully across all testing instances. We consider an individual API call successful if it returns an HTTP 200 status code upon execution. (2) **API Selection Precision (ASP)**: Measures the proportion of predicted API methods that correctly match the ground-truth API endpoint for a given utterance. By disregarding parameter values, ASP isolates the agent’s capability to identify the correct API independent of argument extraction. (3) **Semantic Exact Match (SEM)**: Measures whether an LLM agent generates an API call that semantically matches the ground-truth API call required to answer a user’s utterance. For each evaluated utterance, SEM is assigned a value of 1 if at least one API call generated

by the agent matches the ground-truth API call, and 0 otherwise. Because the agent is permitted up to five iterations to correct formatting errors, invalid parameters or invalid API calls, this metric checks whether the agent successfully converges on the correct API call at any point during its multi-turn reasoning loop. A valid match requires: (1) the same API endpoint; and (2) semantically equivalent parameter values. To compare parameter values, we compute cosine similarity between their embeddings and consider them equivalent when the similarity exceeds 0.95. We selected this threshold empirically by manually inspecting examples with semantic equivalence. We use semantic similarity because several APIs accept free-form textual parameters rather than exact string matches. For example, in a paper search API, queries such as “*large language models for API use*” and “*LLMs for API calling*” correspond to equivalent search intents and should be considered valid matches. We again use Qwen/Qwen3-Embedding-4B [35] to compute parameter embeddings.

4.3.2 Results

We found that **agents achieved lower performance when fulfilling utterances produced by L1-AUG than those generated by P-ALP and ALP, suggesting that utterances simulating linguistic patterns extracted from essays written by English learners are more challenging for LLM agents to fulfill. Furthermore, Qwen-3.5-9B achieved the highest performance.** Table 4 presents the full results.

■ **Table 4** Downstream API calling performance of LLM agents deployed using three LLMs and three benchmarks.

LLM Agent	Method	ESR $\uparrow$	ASP $\uparrow$	SEM $\uparrow$
Qwen-3.5-9B	ALP	0.9482	0.9983	0.8483
	P-ALP	0.9064	0.9908	0.8067
	L1-AUG	0.8870	0.9980	0.6533
Minstral-3-8B	ALP	0.9092	0.9910	0.7100
	P-ALP	0.9033	0.9886	0.6817
	L1-AUG	0.8488	0.9850	0.6583
Llama-3.1-8B	ALP	0.7032	0.9792	0.5650
	P-ALP	0.6886	0.9598	0.5167
	L1-AUG	0.6707	0.9704	0.4783

L1-AUG yielded the lowest ESR and SEM scores for all LLMs. The performance drop from ALP to L1-AUG is most evident on SEM, with drops of 0.1950 for Qwen-3.5-9B, 0.0867 for Llama-3.1-8B, and 0.0517 for Minstral-3-8B, suggesting that utterances incorporating linguistic patterns extracted from essays written by non-native English speakers indeed impact the agent’s ability to extract and map parameters correctly. ASP remained high and stable across all benchmarks and LLMs, ranging from 0.9598 to 0.9983, indicating that API selection is robust to diversity. P-ALP performed between ALP and L1-AUG on most metrics, confirming that persona-based conditioning introduces greater difficulty compared to the baseline, but does not reach the level of challenge posed by L1-AUG.

Qwen-3.5-9B achieved the strongest overall performance, obtaining the highest ESR across all benchmarks and the highest SEM on ALP and P-ALP. Minstral-3-8B ranked second and achieved the highest SEM for L1-AUG, demonstrating competitive performance under the most challenging benchmark. Llama-3.1-8B showed the weakest performance, particularly

on SEM, suggesting higher sensitivity to linguistic diversity in user utterances. Nevertheless, most models showed the same performance ordering across pipelines: ALP achieved the highest scores, followed by P-ALP and L1-AUG. This consistent ordering indicates that the difficulty introduced by L1-aware generation persists across different LLMs.

Table 5 shows utterances from different benchmarks and whether each LLM agent successfully answered them. Example 1 shows a grammatically correct utterance generated by ALP. Ministral-3-8B incorrectly interpreted the temporal clause (“*before the cricket match starts*”) as a mandatory constraint and requested additional information about the cricket match rather than invoking an API. Llama-3.1-8B and Qwen-3.5-9B produced valid API calls and answered the utterance correctly. This highlights how agents may over-interpret information, leading to unnecessary clarification steps. Example 2 illustrates an utterance generated by P-ALP that incorporates additional context given a persona. Llama-3.1-8B issues an invalid API call and responds using its internal knowledge rather than relying on API execution. Ministral-3-8B generates a partially correct response, returning only four exercises instead of six. Qwen-3.5-9B successfully generates both the correct API call and the corresponding final response. Example 3 shows a non-native construction “*movie like legal thriller, for instance A Few Good Men*” that conflates a genre description with a movie title. All three models initially treat “legal thriller” as the movie title, resulting in a failed API call. Llama-3.1-8B repeats the same failing API call three times. Ministral-8B includes the correct movie name on its third attempt and provides the correct answer to the user. Qwen-3.5-9B includes the correct title in its second call but then issues additional API calls for unrelated movies, ultimately failing to provide a final answer.

■ **Table 5** Examples of utterances per pipeline and whether each model successfully resolved the API call. ✓ indicates a correct and complete response, ✓ indicates a partially correct response, and ✗ indicates a failed or incorrect response. Models are Qwen-3.5-9B (Qwen), Ministral-3-8B (Ministral), and Llama-3.1-8B (Llama).

Method	Utterance	QWEN	MINISTRAL	LLAMA
ALP	Give me the current weather in Sydney before the cricket match starts.	✓	✗	✓
P-ALP	Could you find me 6 body-weight exercises for abs? I want a routine that captures raw effort and breath, not another lecture disguised as fitness advice.	✓	✓	✗
L1-AUG	For this weekend,as I am a law student from Myanmar and after reading many case materials,my mind is very tired.So, I need some movie like legal thriller,for instance A Few Good Men, and if there are in 1992, it will be more or less suitable for me.	✗	✓	✗

An interesting observation emerged during evaluation. Some semantically valid API calls returned runtime errors, revealing inconsistencies in the API provider. For example, all agents correctly generated the following API call in response to a request involving Brunei dollars: “name”: “converter_currency”, “arguments”: “amount”: 500, “from_currency”: “USD”, “to_currency”: “BND”. However, this API call returned an execution error because BND was not accepted as the value for the to_currency parameter, despite being listed as a supported currency in the official API documentation. This suggests that benchmarks generated using L1-AUG can expose edge cases in API providers that standard benchmarks would not cover by incorporating country-specific parameter values tied to the user’s nationality. Thus, L1-AUG can serve as a tool for diversity-aware API testing.

4.4 Threats to Validity

In this section, we discuss the main threats to the validity of our findings and the measures taken to mitigate their impact.

Construct Validity. Our quantitative evaluation in RQ1 relies on lexical and syntactic features that capture broad characteristics of writing rather than directly measuring L1 transfer, such as article omission, false cognates, or literal translations. Thus, the quantitative results show that L1-AUG generates utterances with linguistic patterns that differ across countries, but they do not establish that these differences reflect authentic L1 transfer. To address this limitation, we manually analyzed patterns in the generated utterances and compared them with L1 transfer features observed in the reference essays. However, a single annotator conducted this analysis, limiting the reliability of the qualitative assessment.

Furthermore, we extracted the linguistic patterns used to generate utterances from argumentative essays. Consequently, some generated utterances may reflect characteristics of argumentative writing rather than natural interactions with LLM agents. Lastly, L1-AUG generates longer utterances than the baseline methods, which may influence some diversity metrics by providing more opportunities to produce unique words and n-grams. To account for this, we also report TNW to interpret diversity in the context of utterance length.

Internal Validity. Downstream LLM agent evaluation relies on live and third-party APIs. As a result, provider-side failures unrelated to agent performance can affect the ESR metric (e.g., a semantically correct call failing due to a server error). Although all agents were evaluated under the same API conditions, these failures may affect individual evaluations differently depending on the API calls and parameter values each agent generates. Therefore, provider-side failures may introduce noise into comparisons between agents.

External Validity. We focused on generating utterances based on essays from English learners from 20 countries in the ICNALE corpus, 10 APIs, 5 LLMs, and a specific agent configuration. Thus, we cannot generalize the findings to all countries, API domains, LLMs, or agent architectures. Future work should evaluate L1-AUG across these dimensions.

5 Conclusion

In this paper, we introduced L1-AUG, a novel LLM-based pipeline for generating API-calling benchmarks conditioned on linguistic patterns extracted from essays written by non-native English speakers. We also presented an empirical study evaluating LLM agents' ability to fulfill such utterances. Our results show that L1-AUG generates utterances with higher linguistic diversity than alternative generation methods and that these utterances are more challenging for LLM agents to fulfill, exposing weaknesses that other API-calling benchmarks fail to reveal. These findings suggest that incorporating linguistic patterns extracted from real corpora can improve benchmark construction by exposing failure cases that conventional benchmark generation methods overlook. Consequently, L1-AUG can support the robustness evaluation of LLM agents under more linguistically diverse inputs and provide training data for fine-tuning models to better handle variation in non-native English writing. Beyond benchmark generation, L1-AUG can also support diversity-aware API testing by automatically generating diverse API-oriented test cases with diverse parameter values.

This study has limitations that suggest avenues for future research. First, our qualitative evaluation relies on a single human annotator to verify the incorporation of linguistic patterns. Incorporating multiple annotators with L2/L1 transfer expertise in future work would strengthen the reliability of the findings. Second, L1-AUG relies on essay-based corpora

to extract L1 patterns from speakers in 20 Asian countries. Future work should explore additional types of corpora and a broader set of countries to improve linguistic coverage. Third, our benchmark covers ten APIs, and extending it to a large-scale API ecosystem would strengthen the generalisability of our findings. Fourth, we evaluate only off-the-shelf LLMs, and it remains an open question whether models fine-tuned on L1-AUG data would better handle linguistically diverse utterances. Fifth, L1-AUG targets utterances involving a single API call. Future work should extend this to multi-API and multi-turn settings, which more closely reflect the complexity of real-world interactions.

Data Availability Statement

To support reproducibility and transparency, we provide all artifacts associated with this study in an online repository available at: <https://github.com/vitorgaboardi/l1-apg>.

Ethical Considerations

To account for the tendency of LLMs to generate grammatically standard English, our approach includes instructions requiring LLMs to reproduce errors and unusual phrasing only when these are identified in the analyzed essays. Our goal is not to reproduce or reinforce stereotypes associated with linguistic communities, but rather to create diverse benchmarks for evaluating the robustness of LLM agents in understanding utterances from different linguistic backgrounds. The generated utterances should not be interpreted as representative of how all non-native English speakers from a particular country write, nor should L1-AUG be used to generate cultural or linguistic stereotypes.

References

- 1 Poorvi Acharya, J. Elizabeth Liebl, Dhiman Goswami, Kai North, Marcos Zampieri, and Antonios Anastasopoulos. Tracing L1 interference in english learner writing: A longitudinal corpus with error annotations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, 2025*, pages 15146–15167. Association for Computational Linguistics, 2025. doi:10.18653/v1/2025.emnlp-main.766.
- 2 Auday Berro, Vitor Gaboardi Dos Santos, Boualem Benatallah, and Khalid Benabdeslem. Llms to replace crowdsourcing in generating syntactically diverse paraphrases for task-oriented chatbots. In *Advanced Information Systems Engineering - 37th International Conference, CAiSE 2025, Austria*, volume 15701, pages 145–162. Springer, 2025. doi:10.1007/978-3-031-94569-4_9.
- 3 Shoshana Blum-Kulka and Elite Olshtain. Too many words: Length of utterance and pragmatic failure. *Studies in second language acquisition*, 8(2):165–179, 1986. doi:10.1017/S0272263100006069.
- 4 Patricia Salazar Campillo. The use of circumlocution in the foreign language context. *Porta Linguarum: revista internacional de didáctica de las lenguas extranjeras*, 1(5):7–16, 2006.
- 5 Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling synthetic data creation with 1,000,000,000 personas. *CoRR*, abs/2406.20094, 2024. doi:10.48550/arXiv.2406.20094.
- 6 Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL Italy, 2019, Volume 1: Long Papers*, pages 5972–5984. Association for Computational Linguistics, 2019. doi:10.18653/v1/P19-1599.
- 7 Zihao Cheng, Hongru Wang, Zeming Liu, Yuhang Guo, Yuanfang Guo, Yunhong Wang, and Haifeng Wang. ToolSpectrum: Towards personalized tool utilization for large language models. In *Findings of the Association for Computational Linguistics, ACL, Austria*, volume ACL 2025, pages 20679–20699, 2025. doi:10.18653/v1/2025.findings-acl.1063.

- 8 Rena Wei Gao, Xuetong Wu, Tatsuki Kuribayashi, Mingrui Ye, Siya Qi, Carsten Roever, Yuanxing Liu, Zheng Yuan, and Jey Han Lau. Can llms simulate l2-english dialogue? an information-theoretic analysis of l1-dependent biases. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL, Austria*, pages 4355–4379. Association for Computational Linguistics, 2025. doi:10.18653/v1/2025.acl-long.219.
- 9 Yanzhu Guo, Simone Conia, Zelin Zhou, Min Li, Saloni Potdar, and Henry Xiao. Do large language models have an english accent? evaluating and improving the naturalness of multilingual llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL, Austria*, pages 3823–3838, 2025. doi:10.18653/V1/2025.ACL-LONG.193.
- 10 Yanzhu Guo, Guokan Shang, and Chloé Clavel. Benchmarking linguistic diversity of large language models. *Trans. Assoc. Comput. Linguistics*, pages 1507–1526, 2025. doi:10.1162/tacl.a.47.
- 11 Xie He, Arash Habibi Lashkari, Nikhill Vombatkere, and Dilli Prasad Sharma. Authorship attribution methods, challenges, and future research directions: A comprehensive survey. *Inf.*, 15(3):131, 2024. doi:10.3390/info15030131.
- 12 Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyuan Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao Wan, Neil Zhenqiang Gong, and Lichao Sun. Metatool benchmark for large language models: Deciding whether to use tools and which to use. *CoRR*, abs/2310.03128, 2023. doi:10.48550/arXiv.2310.03128.
- 13 Shin'ichiro Ishikawa. *The ICNALE guide: An introduction to a learner corpus study on Asian learners' L2 English*. Taylor & Francis, 2023.
- 14 Rebirra Jemama and Rajesh Kumar. How well do llms imitate human writing style? In *2025 IEEE 16th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 0691–0700. IEEE, 2025. doi:10.1109/UEMCON67449.2025.11267719.
- 15 Khaled Karim and Hossein Nassaji. First language transfer in second language writing: An examination of current research. *Iranian Journal of Language Teaching Research*, 1(1):117–134, 2013.
- 16 Bogdan Kostic, Conor Fallon, Julian Risch, and Alexander Löser. Same meaning, different scores: Lexical and syntactic sensitivity in LLM evaluation. *CoRR*, abs/2602.17316, 2026. doi:10.48550/arXiv.2602.17316.
- 17 James O'Sullivan. Stylometric comparisons of human versus ai-generated creative writing. *Humanit Soc Sci Commun*, 12(1):1–6, 2025. doi:10.1057/s41599-025-05986-3.
- 18 Eileen Pan, Anna Seo Gyeong Choi, Maartje ter Hoeve, Skyler Seto, and Allison Koencke. Analyzing dialectal biases in llms for knowledge and reasoning benchmarks. *CoRR*, abs/2510.00962, 2025. doi:10.48550/arXiv.2510.00962.
- 19 Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): from tool use to agentic evaluation of large language models. In *42th International Conference on Machine Learning, ICML*, volume 267, 2025. URL: <https://proceedings.mlr.press/v267/patil25a.html>.
- 20 Ellie Pavlick and Chris Callison-Burch. Simple PPDB: A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*, pages 143–148. The Association for Computer Linguistics, 2016. doi:10.18653/v1/p16-2024.
- 21 Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, dahai li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations (ICLR)*, pages 9695–9717, 2024.
- 22 Jorge Ramírez, Marcos Báez, Auday Berro, Boualem Benatallah, and Fabio Casati. Crowdsourcing syntactically diverse paraphrases with diversity-aware prompts and workflows. In *Advanced Information Systems Engineering - 34th International Conference, CAiSE 2022, Belgium*, volume 13295, pages 253–269. Springer, 2022. doi:10.1007/978-3-031-07472-1_15.

- 23 Manon Reusens, Philipp Borchert, Jochen De Weerd, and Bart Baesens. Native design bias: Studying the impact of english nativeness on language model performance. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP-AACL*, pages 1195–1215. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics, 2025. doi:10.18653/v1/2025.findings-ijcnlp.73.
- 24 Vitor Gaboardi Dos Santos, Boualem Benatallah, Auday Berro, and Silvana Togneri MacMahon. Diverse utterances generation with GPT to improve task-oriented chatbots and apis integration. In *2nd International Conference on Foundation and Large Language Models, FLLM, United Arab Emirates*, pages 42–50. IEEE, 2024. doi:10.1109/FLLM63129.2024.10852454.
- 25 Raheem Sarwar, Attapol T. Rutherford, Saeed-Ul Hassan, Thanawin Rakthanmanon, and Sarana Nutanong. Native language identification of fluent and advanced non-native writers. *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, 19(4):1–19, 2020. doi:10.1145/3383202.
- 26 Ravindra Singh and Naurang Singh Mangat. Stratified sampling. In *Elements of survey sampling*, volume 15, pages 102–144. Springer, 1996. doi:10.1007/978-94-017-1404-4_5.
- 27 Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, and Le Sun. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. *CoRR*, abs/2306.05301, 2023. doi:10.48550/arXiv.2306.05301.
- 28 Kimberly Le Truong, Riccardo Fogliato, Hoda Heidari, and Steven Wu. Persona-augmented benchmarking: Evaluating llms across diverse writing styles. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP, China*, pages 22676–22709. Association for Computational Linguistics, 2025. doi:10.18653/v1/2025.emnlp-main.1155.
- 29 Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers Comput. Sci.*, 18(6):186345, 2024. doi:10.1007/s11704-024-40231-1.
- 30 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems, NeurIPS 2022, USA*, volume 35, pages 24824–24837, 2022. doi:10.52202/068431-1800.
- 31 Han Xu, Ruining Zhao, Jindong Wang, and Haipeng Chen. RESTful-llama: Connecting user queries to RESTful APIs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: EMNLP - Industry Track, USA*, pages 1433–1443. Association for Computational Linguistics, 2024. doi:10.18653/v1/2024.emnlp-industry.105.
- 32 Weikai Xu, Chengrui Huang, Shen Gao, and Shuo Shang. Llm-based agents for tool learning: A survey. *Data Sci. Eng.*, 10(4):533–563, 2025. doi:10.1007/s41019-025-00296-9.
- 33 Mohammad-Ali Yaghoub-Zadeh-Fard, Boualem Benatallah, Fabio Casati, Moshe Chai Barukh, and Shayan Zamanirad. User utterance acquisition for training task-oriented bots: A review of challenges, techniques and opportunities. *IEEE Internet Comput.*, 24(3):30–38, 2020. doi:10.1109/MIC.2020.2978157.
- 34 Shengcheng Yu, Yuchen Ling, Chunrong Fang, Zhenyu Chen, and Chunyang Chen. Towards automated crowdsourced testing via personified-llm. *CoRR*, abs/2603.24160, 2026. doi:10.48550/arXiv.2603.24160.
- 35 Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *CoRR*, abs/2506.05176, 2025. doi:10.48550/arXiv.2506.05176.
- 36 Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Taxygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100, 2018. doi:10.1145/3209978.3210080.