Multi-Layered Legislative Knowledge Management with Property Graphs

Andrea Colombo
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy

Francesco Cambria
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy

Francesco Invernici
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy

Abstract

The sparse nature and intricate set of relationships between legislative acts pose a significant challenge in the choice of the underlying database model, which both allows for performing structured queries and developing intuitive and smooth knowledge management. In this paper, we propose to use Property Graphs as a powerful alternative for managing legislative knowledge. First, we discuss how graph queries are a valid alternative solution to standard legislative knowledge management by showing how our data model fully captures the problem of law versioning (i.e., the existence of many versions for the same law). Then, we analyze, propose and implement innovative ways for monitoring the legislative system using Property Graph tools that have been recently standardized and developed, such as triggers and graph-based association rules, which empower our model of advanced ways of handling legislative data. For instance, we will show how we can use these tools to develop intelligent warning systems that inform stakeholders of critical changes in legislation through active rule reasoning or to detect shifts in graph patterns via continuous monitoring of significant association patterns. Lastly, we will present an approach to expand the Property Graph model to also include laws from an institutional lower layer: regional governments. We then show how this would open new ways to exploit Property Graph tools for legislative knowledge management.

2012 ACM Subject Classification Applied computing → Law; Information systems → Graph-based database models; Information systems → Triggers and rules

Keywords and phrases data management, property graphs, triggers, law

Digital Object Identifier 10.4230/TGDK.4.2.4

Category Research

Related Version *Previous Version*: `https://edbticdt2025.upc.edu/files/TGD/TGD-1.pdf`

Supplementary Material *Dataset*: `https://zenodo.org/records/13798158` [14]
Software: `https://github.com/FrancescoCambria/Cypher-Queries-TGDK26` [9]
archived at `swh:1:dir:c175195a74440b097ec11f0fb49fdc67cf616930`

Received 2025-07-15 **Accepted** 2026-06-22 **Published** 2026-09-03

Editors Stefania Dumbrava, George Fletcher, Matteo Lissandrini, and Riccardo Tommasini

Special Issue Data Management for (Knowledge) Graphs

Generative AI Declaration Generative AI was used during the preparation of this manuscript exclusively for grammatical refinement and literature discovery. The authors carefully reviewed all AI-assisted outputs and assume full responsibility for the accuracy of the final work.

1 Introduction

Legislative data comprises the set of acts, bills, or other normative documents produced by an authority that holds legislative power to regulate some domains on a national, regional, or international level. In most democratic countries, such power is vested in parliaments, whether

© Andrea Colombo, Francesco Cambria, and Francesco Invernici;
licensed under Creative Commons License CC-BY 4.0

national, regional, or federal, depending on the country's political system. One of the main challenges when dealing with such textual documents is converting them into machine-readable formats that enable adopting structured approaches when querying and analysing legislative acts.

To tackle this, the computer law community has devoted many efforts to proposing appropriate international standards to capture the common grounds of laws enacted in different legislative systems. Most of previous works have been based on the eXtensible Markup Language (XML) format, a semi-structured data model that has been naturally used for representing textual and hierarchical data, such as laws [32]. Relevant XML-based proposals include the Legal Knowledge Interchange Format (LKIF) [26], LegalRuleML [3] and Akoma Ntoso [4].

However, the need for a user-friendly storing of textual documents (with XML tags and hierarchical structures being ideal for such aims) collides with the possibility of conducting quantitative analysis of the complexities and features of the legislative system [30], which strongly depends on the ability to traverse relationships (e.g., dependencies and references between laws and articles). In fact, while the creation of XML formats to represent the components of an individual law (i.e., article or section structure, tags for references to other laws) is extremely useful for structured access to a document, it becomes cumbersome to manage and analyze trends and the complexity of the legislation. Furthermore, while XML-native databases generally support tools such as triggers [8], their focus on the document-level logic (e.g., handling events as document create, document update, document delete, property changes) harms the potential of *reasoning* over legislative data, leveraging relationships among acts to discover and detect anomalous patterns that require attention. For instance, abrogations of some portions of legislation can inadvertently create voids in the legal foundations of third acts. Thus, a data model that prioritizes relationships among acts is crucial in monitoring such scenarios.

In this paper, we present and discuss an innovative approach to legislative knowledge management based on Property Graphs (PG). By leveraging state-of-the-art advancements from the database community in the formal standardization of Property Graphs and their tools [2, 11, 10], we present an alternative approach for handling legislative acts, which maintains the ability to manage documents as hierarchical structures, with the creation of dedicated graph relationships, but that enriches it with tools of additional utility. First, we demonstrate how our proposal seamlessly handles textual documents by focusing on one of the main issues of legislative knowledge management, that is, the law versioning problem, i.e., retrieving the text of a law in force in a certain timestamp, thus accounting for successive modifications and/or abrogations.

Then, we show how, by explicitly modelling references among acts as graph relationships, we enable more sophisticated tools to analyze and monitor trends in the legislative activity [17], such as by the development of intelligent systems that can *reason* over legislative data through the use of active rules in the form of Property Graph triggers [11]. To this aim, we develop a trigger-based warning system that monitors the legislation, capable of detecting harmful scenarios within the legislative system, such as the creation of legislative voids. We demonstrate how we capture the latter scenario by using active rules in the form of triggers expressed in Cypher, the closest language compliant with the Graph Query Language standard. We demonstrate the effectiveness of this warning system over the Italian legislation, for which we have implemented our PG-based schema and stored it within a Neo4j database. We evaluate how warnings signaled by our reasoning-based approach significantly positively correlate with ex-post-legislative interventions that aim at fixing the legislative void, demonstrating their predictive power for a timely detection of legislative voids.

Additionally, we examine how association rules can serve as a valuable tool for knowledge discovery, offering insights into trends in the evolution of the legislative system. To this end, by leveraging the recently developed association rule operator for property graphs [10], we present an application in which association rules can aid in detecting the attitude of governments towards a topic of particular interest, i.e., whether there is a discontinuity among successive governments. We focus on legislation related to "procurement", and we find that, in Italy, there is a continuous significant stream of creations and abrogations of rules governing this topic, indicating a certain degree of instability in the topic in recent years.

In this paper, we aim to exploit the dual nature of the Property Graph model, which combines a network structure with rich data properties. Unlike traditional legislative models that prioritize the hierarchical organization of documents, our approach leverages both the relational connections between legal acts (the graph) and their textual content and metadata (the properties), allowing a flexible representation of both content and, as we will see, the inter-level connection among laws.

In addition, we demonstrate that tools native to Property Graphs effectively address legislative management challenges. This model provides an effective way to store and model legislative texts, leveraging the graph structure to enable better querying and reasoning over legal dependencies and identify behavioral patterns.

We leverage the flexibility of the Property Graph model to extend the existing representation of legislative knowledge by incorporating laws issued by lower institutional levels, specifically regional governments. While national legislation is often centralized and relatively uniform, regional laws introduce additional complexity due to the heterogeneous ways in which each regional government stores and publishes legal documents. In particular, we focus on integrating legal data from two regions: Lombardia and Liguria. Each region adopts different formats and publication mechanisms, which require dedicated processing pipelines. By adding this new regional layer to the Italian Legislative Graph, we not only enrich the overall structure of the graph but also introduce new opportunities for advanced legal knowledge analysis, which we later illustrate with an example.

In this work, we address three main research questions that highlight the effectiveness of Property Graphs for legislative data:

RQ1: How can the Property Graph model be effectively exploited to manage the evolution of legislative acts (law versioning)?
RQ2: How can recently developed PG tools like Cypher-like triggers and association rules mining serve legal experts in their work?
RQ3: How can the Italian legislative graph be extended to effectively represent different *levels* of legislation within the same graph?

Overview. The remainder of the paper is organized as follows. In Section 2 the Related Work is introduced. Section 3 presents the Property Graph Schema that we adopt for modelling legislative knowledge, discussing its implementation in the Italian system and how it can retrieve time-dependent versions of laws. In Section 4, we present how tools like PG triggers and graph association rules can be used to facilitate knowledge discovery and automated monitoring systems. In Section 5, we perform our experiments to demonstrate the benefits of using a graph database and its rich set of tools to manage legislative knowledge. Then, in Section 6, we describe the necessary steps for adding regional laws to the Italian graph and we show how this new layer of data can be exploited for legislative knowledge management.

2 Related Work

The Property Graph (PG) data model has gained significant traction for managing complex, interconnected data, serving as a flexible alternative to relational models. In [2], Angles et al. proposed PG-Schema, a standardized language for defining graph structures, which allows for the rigorous validation of node and edge types within a property graph. In addition, a new standard for GQL, the emerging query language for property graph databases, has been proposed in [25].

Property graphs have been enriched with active data management tools. In [11], we proposed a formalization of triggers, originally a staple of relational database systems, for property graphs. The PG-Triggers enable reactive behaviors, such as automatically detecting complex events or maintaining consistency, directly within the graph storage. In [10], instead, Cambria et al. present the MINE GRAPH RULE operator to adapt association rules mining from transactional databases to property graphs.

While legislative data has traditionally been managed as static documents [26, 3, 4], recent approaches have demonstrated the utility of Property Graphs for capturing the dynamic nature of legal systems. In [19], we proposed the first approach for representing the Italian legislative system, in terms of documents and references, through a Property Graph, together with its schema [2]. While our proposal was country-specific, its foundation relies on an internationally adopted standard (OASIS) [36], which aimed to define common elements that characterize legislative acts across multiple legislative traditions [46]. For instance, in [20] we constructed the graph reflecting the legislation of the United States, modeled around the same graph schema.

Other than Property Graphs, RDF (Resource Description Framework) and Knowledge Graphs based on Semantic Web standards have been used to manage legislative knowledge exploiting graphs. For instance, in [37] Oliveira et Oliveira propose an RDF model for a legislative system and apply it on the Brazilian legislation. In [43], the authors propose an approach based on a Knowledge Graph to model legal documents so that the legal expert can design queries to easily retrieve relevant information.

3 Property Graphs for Legislative Data

Modelling legislative data and the interconnections between laws and articles in a property graph offers a structured and dynamic approach to understanding and managing the knowledge of a legislative system.

In this section, we present a Property Graph Schema that aims to handle legislative data and discuss how we can leverage the expressiveness of a graph query language, such as Cypher, the closest GQL-compliant language, to handle a fundamental feature in legislative knowledge management: the retrieval of the correct version of a law based on a desired timestamp.

3.1 Property Graph Schema

In Figure 1, we visually present the schema we proposed in [19] for the Italian legislation. In a graph, nodes can represent laws connected to their articles (or, according to the legislative tradition, sections, clauses, etc.) through a parthood relationship. Metadata relevant to the law, such as title, publication date, and law domain (indicating the ministries responsible for the law's content), are assigned as properties directly to the law node, enabling a structured approach to query and analyze the legislative corpus. Articles are distinct nodes with properties that detail the specific sections of the law they represent. These properties include the article text, any headings or titles specific to the article, and a list of policy-related topics that the article governs. Attachments, which serve a different function within a law document than articles, are also modelled as distinct nodes with their characteristics. Law, article and attachment nodes are interconnected using a set of directed edges that capture references, dependencies, and amendments that denote the evolution of a legislative system. We included four fundamental types of relationships which we identified as generic enough across multiple legislative traditions: an *"is the legal basis of"* dependence, an *amends*, an *abrogates* and a more generic *cites* reference. The classification of references in the schema categories follows the official annotations present in the original sources. Such a dataset of

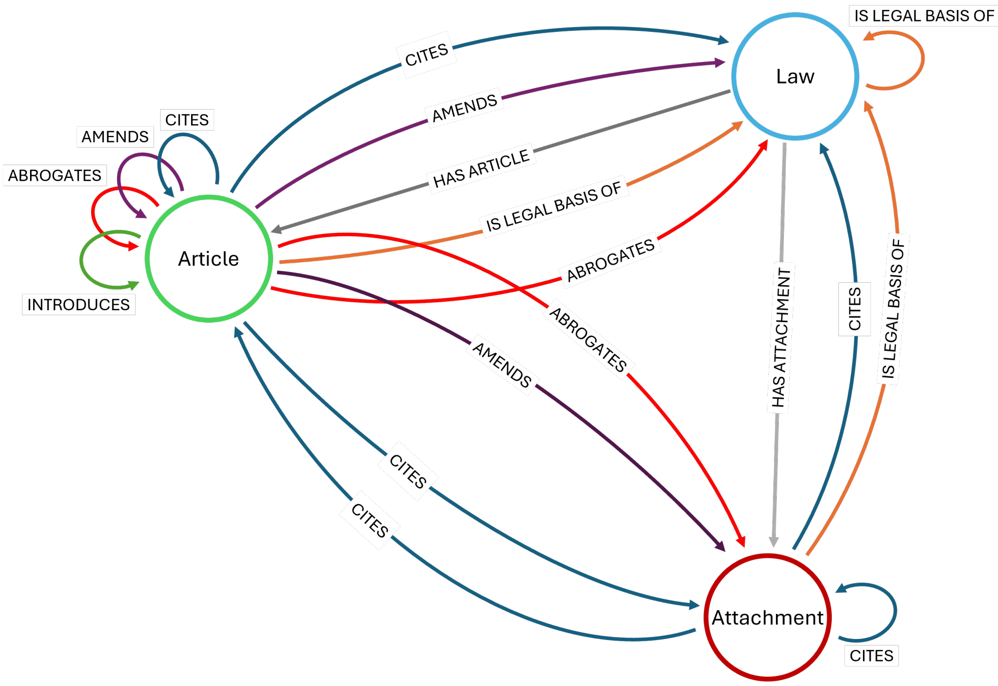

Figure 1 Graphical visualization of the Property Graph Schema. According to this schema, legislation can be represented by three node labels, `(:Law)`, `(:Article)`, and `(:Attachment)`. Laws can be the legal basis with each other, or have articles and attachments. Articles and attachments can cite each other or laws, and be the legal basis of laws. Finally, articles can also introduce other articles, or amend and abrogate laws, attachments, or other articles. – *Properties are omitted for better visualization.*

official annotation can be used to train reference classifiers, based on NLP models as BERT, to classify edges in scenarios where the annotation is not available (i.e., in the regional layer, as we will see).

The formal PG-Schema, as defined in [2], that we use to model legislative data is depicted in Figure 2.

Regarding edge properties, all defined types of references share the common attribute *paragraph*, which denotes, if applicable, the specific portion of the article's text that is being referenced. For instance, one could be interested in citing or amending only a specific portion of text within the article. Thus, such a feature can be captured by inserting the paragraph of interest. As a consequence of such a modelling choice, an *abrogates* edge whose *paragraph* property is `NULL` implies a full abrogation of the destination article. Such a modeling choice follows the empirical observation that a complete repeal of an article never occurs by repealing all its paragraphs on an individual basis; this means that, a complete repeal occurs regardless of the specific paragraph previously deleted. A law becomes abrogated whenever all its articles have been abrogated. Thus, a law abrogation can be inferred by looking at whether all its articles have an `ABROGATES` edge without the *paragraph* property (which would denote a partial abrogation).[1]

Finally, *amends* and *abrogates* edges are the only types of references that also share a second property: *newText*. Such (textual) property captures the new text of the destination article node, as modified/abrogated by the source article node.

[1] Abrogation edges, according to our model, can theoretically also be global repeals, deleting an entire law (i.e., a `Law` node is the destination node). However, a global repeal can be equally achieved by creating multiple abrogation edges which delete each article of the destination law. Consequently, in the rest of the paper, we will only consider the scenario of abrogation edges with an Article node as destination.

```
CREATE GRAPH TYPE lawsGraphType STRICT{(lawType: Law {id STRING, title STRING,
   lawNum INT, typeLaw STRING, publicationDate DATE, numArt INT, numAttach INT}),
  (articleType: Article {id STRING, title STRING, number INT}),
  (attachmentType: Attachment {id STRING, title STRING, type STRING}),
  (:lawType)-[hasArticleType: has_article]->(:articleType),
  (:lawType)-[hasAttachmentType:has_attachment]->(:attachmentType),
  (:lawType)-[referenceType: is_legal_basis_of]->(:lawType),
  (:articleType)-[referenceType:is_legal_basis_of|amends|abrogates|cites]->(:lawType),
  (:articleType)-[referenceType: amends|abrogates|cites
  {paragraph STRING, newText STRING}]->(:articleType),
  (:articleType)-[referenceType:introduces]->(:articleType),
  (:articleType)-[referenceType: amends|abrogates
   {paragraph STRING, newText STRING}]->(:attachmentType),
  (:attachmentType)-[referenceType:is_legal_basis_of|cites]->(:lawType),
  (:attachmentType)-[referenceType:cites{paragraph STRING}]->(:articleType),
  (:attachmentType)-[referenceType:cites {paragraph STRING}]->(:attachmentType)}
```

Figure 2 PG-schema of the legislative data model proposed.

The Italian Legislative Property Graph. We implemented our schema for the Italian legislation and derived a graph consisting of Italian national laws. The graph is stored in a Neo4j database (available at [16]), comprising 74k law nodes, 318k article nodes, 127k attachment nodes, 107k legal basis dependencies, 64k abrogations, 80k amendments, and 228k generic citations. The graph is actively maintained, loading any new laws or modifications daily. For the aim of this paper, we will use acts and laws as interchangeable terms to refer to the official normative documents produced by the Italian institutions. Articles represent portions of the law instead, following the structure of the law. Attachments represent a particular type of articles, e.g., tables, figures or other annexes that are part of the law.

As discussed in [19], the quality of the graph of Italian legislation was evaluated using a multidimensional framework that focused on accuracy, completeness, and consistency. The ETL pipeline achieved high accuracy, correctly inferring 98% of true law abrogations, as demonstrated by comparing the legacy system (the official portal of Italian legislation, which stores updated versions of laws). The high consistency was evaluated by detecting material errors in edge classification using text diffs as a heuristic to classify edge types, which accounted for only 1.8% of total edges.

3.2 Managing Law Versioning via Graph Queries

Legislative systems evolve over time, generating a continuous stream of new laws that amend or repeal existing ones. Each change represents a new version of the law, capturing the updated text whenever a modification occurs. This process often results in rapidly growing use of textual documents, as even minor changes necessitate the storage of additional files with the novel text.

By adopting the graph model and the schema that we proposed, we can seamlessly handle such features by (i) storing the original textual version of each article of law within a node property, (ii) storing the modified textual version, as per amended by successive laws, as an edge property and (iii) delegating the computation of the in-force text for a law at a certain timestamp to a graph query, that, by leveraging publication dates available in law nodes, is capable of always infering the desired law version. By doing so, we avoid storing multiple textual versions of the same law, achieving a more efficient storage of legislative data and, at the same time, allowing a more structured and straightforward approach for querying statistics or detecting trends of a legislative system.

Text of articles of the law of interest in their latest modified version before the desired timestamp

```
CALL {
  MATCH (l:Law)-[:HAS_ARTICLE]→(a:Article)
  OPTIONAL MATCH (a)←[r:ABROGATES|AMENDS|INTRODUCES]-(a2)←[:HAS_ARTICLE]-(l2:Law)
  WHERE l2.publicationDate < datetime({$TIMESTAMP})
  WITH l.id AS IDLAW, a.id ASIDART, a.number AS NUMART, MAX(l2.publicationDate) AS LASTMOD
  WHERE IDLAW = {$IDLAW}
  WITH IDLAW, IDART, NUMART, LASTMOD
  MATCH (l:Law)-[HAS_ARTICLE]→(a:Article)←[r:ABROGATES|AMENDS|INTRODUCES]-(a2)←[HAS_ARTICLE]-(l2:Law)
  WHERE l.id = IDLAW AND a.id = IDART AND LASTMOD = l2.publicationDate
  RETURN IDLAW, IDART, r.newtext AS TEXT, NUMART

  UNION
```

Text of articles of the law of interest which have modifications only after the desired timestamp

```
  MATCH (l:Law)-[HAS_ARTICLE]→(a:Article)←[r:ABROGATES|AMENDS|INTRODUCES]-(a2)←[HAS_ARTICLE]-(l2:Law)
  WITH l.id AS IDLAW, a.id AS IDART, COUNT(r) AS NCHANGES
  WHERE IDLAW = {$IDLAW}
  WITH IDLAW, IDART, NCHANGES
  MATCH (l:Law)-[HAS_ARTICLE]→(a:Article)←[r:ABROGATES|AMENDS|INTRODUCES]-(a2)←[HAS_ARTICLE]-(l2:Law)
  WHERE a.id = IDART AND l2.publicationDate >= datetime({$TIMESTAMP})
  WITH IDLAW, IDART, a.number AS NUMART, a.text AS TEXT, NCHANGES, COUNT(*) AS FUTURECHANGES
  WHERE NCHANGES = FUTURECHANGES
  RETURN IDLAW, IDART, TEXT, NUMART

  UNION
```

Text of articles of the law of interest that have never been amended

```
  MATCH (l:Law)-[HAS_ARTICLE]→(a:Article)
  WHERE NOT (a)←[:ABROGATES|AMENDS|INTRODUCES]-() AND l.id ={$IDLAW}
  RETURN l.id AS IDLAW, a.id AS IDART, a.text AS TEXT, a.number AS NUMART
}
WITH TEXT, NUMART
WHERE TEXT IS NOT NULL
WITH TEXT, NUMART
ORDER BY NUMART ASC
RETURN COLLECT(TEXT)
```

In-force text of desired law at date T

Figure 3 Annotated Cypher query deriving the in-force text of law `IDLAW` at a certain `TIMESTAMP`. In red, the two variables to be set according to the user's requests. First, a subquery (within the `CALL` construct) is used to derive and union the law articles in-force at the desired timestamp. Then, a `COLLECT` function is used to produce the correct law text as a document.

In Figure 3, we present the query in Cypher that we designed to manage law versioning based on our proposed schema. Such a query strongly relies on navigating graph patterns to combine temporal information. In detail, by specifying a `TIMESTAMP` T and a law of interest L (i.e., an `IDLAW`), we can get the text version of L at T by performing a union that concatenates articles of L in their latest modified version, as per modified by the law with the maximum publication date before T, with articles that have never been amended or have been amended only after T, thus do not have any incoming edge of types `ABROGATES` and `AMENDS`.

Finally, we note how this approach is replicable even on a more granular level, i.e., by considering paragraphs as graph nodes. The benefit would be even more pronounced since most of the modifications change only some words of a law, and a standard document versioning approach would require the storage of an additional document, even for tiny substitutions. Here, the challenge would be consistently modelling paragraph nodes and linking textual references to such nodes, since references do not always explicitly mention the paragraph, as we experienced with the Italian legislation.

In Figure 4, we computed the query execution times of law versions based on the number of articles to evaluate the scalability of our versioning approach. The results indicate that the execution time remains consistently low, with the median duration for all bins staying within a narrow range of approximately 0.75s to 1.1s. Although we observe a higher variance in the 21–25 article bin – likely due to specific document complexities or system cold starts, there is no discernible trend in execution time as the number of articles increases. This stability suggests that the query performance is largely independent of the document size within the tested parameters, confirming that our indexing strategy effectively mitigates the overhead typically associated with larger legal datasets.

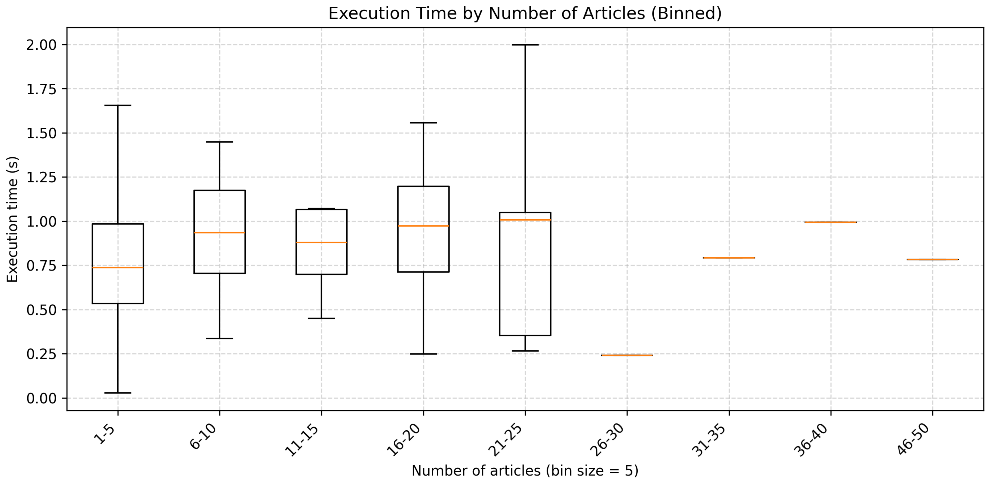

Figure 4 Execution time of the versioning query (Figure 3) grouped by the number of articles, from a random sample of 250 laws. The median execution time remains relatively constant (under 1.25s) across all bins, suggesting that query performance scales well regardless of the article count.

```
(l1:Law)-[:HAS_ARTICLE]->(a1:Article)<-[r:ABROGATES|AMENDS|INTRODUCES]-(a2:Article)
    <-[:HAS_ARTICLE]-(l2:Law)
```

Figure 5 Path for amending laws to modified articles.

3.3 Benefits of a Legislative Property Graph Modeling

In this section, we examine the practical motivations for modeling legislative data as a property graph rather than a traditional relational schema. To illustrate the advantages, consider Figure 3, which depicts a core operation of a legislative database: computing updated text versions while maintaining a historical record of modifications. In our model, this is efficiently handled through graph patterns, as shown in Figure 5.

This pattern directly connects an amending law to the modified law via its articles. While various relational schemas could be implemented to represent this (also in a more compact form, which, however, might lose some other features), a direct SQL translation approach of our schema would be significantly more verbose and computationally intensive. In a relational system, traversing these relationships typically requires navigating multiple associative (many-to-many) tables or complex self-referencing foreign keys. Furthermore, because each relationship type (`ABROGATES`, `AMENDS`, and `INTRODUCES`) has some unique properties, thus a normalized relational model would likely require dedicated join tables for each, further increasing query complexity.

Beyond performance, the graph approach offers superior semantic readability. In a property graph, the query syntax visually maps to the conceptual domain model. This reduces the cognitive gap between a legislative researcher's question and the technical code required to answer it.

4 Triggers and Association Rules

Triggers have been a fundamental feature since the inception of relational databases [24]; they were explored in depth in [47] and formalized in the ISO-ANSI SQL3 Standard [35]. Traditionally, triggers have been utilized to automate tasks that facilitate the development of automation systems [41]. In graph databases, adding reactive components has also been helpful in supporting important applications, such as the monitoring of pandemic events [11], which naturally are modeled into knowledge graphs stored in graph databases. Formalizations of triggers for graph databases and Property Graphs have recently been proposed [11, 5]. Due to the richness of the graph data model w.r.t. the relational model, triggers can also be leveraged to encode more complex (and interesting) events. For instance, since property graph items can be either nodes or relationships, triggers can include the creation and deletion of nodes/relationships as well as the setting and removal of their labels and properties. The strong analogy between triggers, i.e., active rules, and deductive rules, as defined in [12], has also enabled the use of triggers for adopting *reasoning* approaches over data modelled into knowledge graphs stored as property graphs.

Association rules (AR) mining was first introduced in [1, 44] and aim to uncover patterns or correlations within data. Classically, association rules take the form of $X \rightarrow Y$ where X is referred to as the body of the rule and Y as its head. Both body and head represent predicates or conditions that can be validated within the data, based on two key metrics: support and confidence. Association rules mining has been applied in many different domains: for instance, for highlighting which items are frequently purchased together [13, 38], for detecting connections between symptoms and diseases in medicine [45], or to uncover patterns of user interest in social networks [42]. Like triggers, the flexibility and richness of the graph data model allow association rule mining to uncover more insightful patterns within the data. The MINE GRAPH RULE operator, introduced in [10], leverages all the capabilities of the graph query language to simplify the definition and extraction of association rules from graph databases. While the operator is not fully integrated into the standard Cypher language, its syntax closely follows Cypher, ensuring clarity and readability. Furthermore, the queries used to extract association rules can be implemented in any graph database management system that supports GQL.

In the rest of this section, we present two practical applications of such tools, demonstrating their contribution to implementing more sophisticated monitoring activities in the legislative domain.

4.1 Warnings for Legal Basis Abrogation

In the legislative context, reasoning through active rules can be used to create automatic monitoring systems that derive warnings for anomalous events based on graph patterns. We identified an important application of such a feature in monitoring events related to repealing key acts that serve to form a *legal basis* for other pieces of legislation. Such abrogations might create legislative voids and hinder the applicability of third laws, which have their foundation in the repealed law. In other words, the content and rules stated in an act might depend on legislation abrogated by the newly enacted law.

In our schema, we modelled legislative dependencies through `IS LEGAL BASIS OF` edges, which define the foundations of a given law. Thus, to implement the application, we can utilize the graph structure to generate timely warnings about potentially hazardous situations that require monitoring. Such warnings take the form of new special edges that can be visually inspected. In detail, we can define the following deductive rules that generate warnings for this problem:

1. Whenever a new law containing abrogations is published, we *compute* whether the new law has repealed an older law by leveraging the graph schema, as discussed in Section 3. In such cases, we *merge* `REPEAL` edges that connect the new law node with the abrogated law nodes.
2. By traversing the `IS LEGAL BASIS OF` edges, a second reasoning rule monitors whether the `REPEAL` of a law has created a legislative void in the legal foundation of a third in-force law. For such cases, it creates a warning edge that informs about the potentially harmful situation (which we denote as a `UNDERMINES` edge).

A visual example of the reasoning process is available in Figure 6.

Algorithm 1 Legislative Warning Generation.

Input: $\mathcal{G} = (V, E)$ (Graph), L_{new} (New Law)
Output: $\mathcal{G}' = (V, E')$ (Updated Graph with new `REPEAL` and `UNDERMINES` edges)
// Phase 1: Aggregate Article Abrogations into Law Repeals
$a := article$
$\mathcal{L}_{target} = \{L | (L) \xrightarrow{HAS\_ARTICLE} (a) \wedge (L_{new}) \xrightarrow{ABROGATES} (a)\}$
1: **for each** L_{target} **in** $\mathcal{L}_{target}$ **do**
2: $A_{total} \leftarrow$ count a where $(L_{target}) \xrightarrow{HAS\_ARTICLE} (a)$
3: $A_{abrogated} \leftarrow$ count a where $(L_{new}) \xrightarrow{ABROGATES} (a)$
4: **if** $A_{abrogated} \geq A_{total}$ **then**
5: **merge** $(L_{new}) \xrightarrow{REPEAL} (L_{target})$
6: **end if**
7: **end for**
// Phase 2: Detect Impacts on Dependent Legislation
$e := edge$
$\mathcal{E} = \{\forall e | e : (L_{new}) \xrightarrow{REPEAL} (L_{target})\}$
8: **for each** e **in** $\mathcal{E}$ **do**
$\mathcal{L}_{dep} = \{\forall L_{dep} | (L_{target}) \xrightarrow{IS\_LEGAL\_BASIS\_OF} (L_{dep})\}$
9: **for each** L_{dep} **in** $\mathcal{L}_{dep}$ **do**
10: **if** $\nexists L \xrightarrow{REPEAL} L_{dep}$ **then**
11: **merge** $(L_{new}) \xrightarrow{UNDERMINES} (L_{dep})$
12: **end if**
13: **end for**
14: **end for**

The logic described in the pseudocode in Algorithm 1 is implemented via Cypher triggers. The first phase (line 1 to 7) monitors the insertion of new `ABROGATES` edges; for each of these edges, we count the total distinct number of abrogations to each article of a law (line 2, considering only edges whose *paragraph* property is null (denoting full article abrogation). When all articles have been abrogated, then a new `REPEAL` edge is created (line $4-6$). The second phase captures the newly created `REPEAL` edges and checks for dependent laws (L_{dep}) (lines $8-9$) that lack repealing edges (line 10): for such cases, a new `UNDERMINES` edge is merged, highlighting the possibility of the creating a potential harmful legislative void.

More in detail, the first rule can be encoded within the trigger expressed in Cypher in Figure 7, a possible implementation of PG-Triggers, as defined in [11].

In the PG-trigger, the `UNWIND` clause captures, within the Neo4j ecosystem, the insertion of a new `ABROGATES` edges which derive from the insertion of a new law node. If a graph pattern denoting an article abrogation without the *paragraph* property exists (denoting, according to our schema, a full abrogation), then the full abrogation is tested by monitoring whether all its articles have been repealed. In such a scenario, the trigger *merges* a new edge denoting an abrogation (i.e., a `REPEAL`) of the destination law node.

The second rule can be implemented as shown in Figure 8, where, here, the `UNWIND` clause captures the newly created `REPEAL` edges and checks whether there exists a non-abrogated law whose the repealed one is the legal basis of. Within the `WHERE` conditions, we exclude the cases in which the repealing law already contains any explicit "correction" to the destination law and its articles. If such a pattern exists, we merge new derived relationships, i.e., `UNDERMINES` edges, which denote a potential legislative void since the newly published law is deleting a law that is

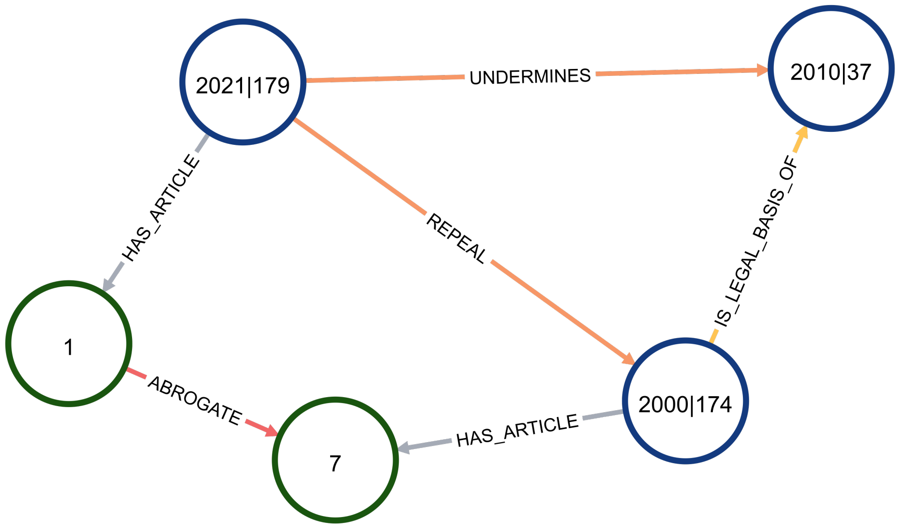

Figure 6 Law 2021/179 repeals law 2000/174, by abrogating its remaining in-force article (i.e., article 7). However, since law 2000/174 is a legal foundation of law 2010/37, law 2021/179 also undermines its validity since its foundation has been removed, raising a potential legislative void.

```
UNWIND $createdRelationship AS newRel
MATCH p=(law:Law)-[:HAS_ARTICLE]->(art:Article)<-[newRel:ABROGATES]-(art2:Article)
               <-[:HAS_ARTICLE]-(newLaw:Law)
WHERE newRel.paragraph IS NULL
WITH law, art, newLaw
MATCH (law)-[:HAS_ARTICLE]->(art)<-[abr:ABROGATES]-()
WHERE abr.paragraph IS NULL
WITH law, law.numArt AS NUMART, COUNT(DISTINCT art) AS NUMREPEALS, newLaw
WHERE NUMREPEALS >= NUMART
MERGE (newLaw)-[rep:REPEAL]->(law)
```

Figure 7 PG-Trigger encoding the first rule.

used as a legal foundation in a third law. Thus, such edges require attention from the legislator and any interested stakeholders in the domain of the law. The warnings might translate into corrective actions, i.e., new laws or articles that address potential legislative voids.

These triggers can be executed asynchronously, thus ensuring that the monitoring system does not introduce significant overhead during the ingestion of large legislative datasets. This reactive approach transforms the graph from a passive repository of legal texts into an active decision-support system. Specifically, the emergence of an `UNDERMINES` edge serves as a semantic alert for legal experts. Instead of manually auditing the cascading effects of a repeal – a task that grows more complex with the size of the corpus – legislators can query for these specific patterns to prioritize their interventions.

```
UNWIND $createdRelationships AS newRel
MATCH (art)<-[:HAS_ARTICLE]-(newLaw:Law)-[newRel:REPEAL]->(law:Law)
          -[:IS_LEGAL_BASIS_OF]->(law2:Law)-[:HAS_ARTICLE]->(art2:Article)
WHERE NOT EXISTS (()-[:REPEAL]->(law2))
AND newLaw.id <> law2.id
AND NOT EXISTS ((art)-[:ABROGATES|CITES|AMENDS|INTRODUCES]->(art2))
MERGE (newLaw)-[:UNDERMINES]->(law2)
```

Figure 8 PG-Trigger encoding the second rule.

```
MINE GRAPH RULE FrequentChangesOnTopic
GROUPING ON (topic:Topic)
WHERE t.name CONTAINS '$topicName'
DEFINING
    BODY AS (topic)<-[:OF_TOPIC]-(art:Article)<-[:HAS_ARTICLE]-(law:Law)
               <-[:CREATED_UNDER]-(gov:Government)-[:SUCCEEDED_BY]->(gov2.Government)
    HEAD AS (topic)<-[:OF_TOPIC]-(art2:Article) <-[:ABROGATES]-(art3:Article)
               <-[:HAS_ARTICLE]-(law2:Law)<-[:CREATED_UNDER]-(gov3:Government)
WHERE gov2 = gov3
IGNORE art, law, gov2, art2, art3, law2
EXTRACTING RULES WITH SUPPORT > 0.2
               AND CONFIDENCE > 0.2
```

Figure 9 Mine Graph Rule operator for extracting governments behavioural patterns.

4.2 Patterns of Government Attitude

In a legislative graph proposed in Section 3, items represent laws, articles, and attachments. Such a schema can be easily enriched by adding governments and topic nodes[2]. The former represents the governments under which laws have been enacted; the latter connects each legislative node to the topics its content refers to. With these enrichments, we can exploit (graph) association rules to understand which and if governments have the attitude to profoundly change legislation referring to a topic for which its predecessor government had also previously implemented legislation. Through this application, we can detect *temporal graph patterns* that monitor the behaviour of governments and identify discontinuity evolution of the legislation. The mining of this kind of pattern can be achieved by adopting the `MINE GRAPH RULE` operator, whose syntax was proposed in [10], and can be written as shown in Figure 9. In [10], the operator's performance was also evaluated, proving that it can extract rules reasonably fast on graphs with similar dimensions. For this reason, we chose to exploit it for deeper analysis of the Italian Legislation graph.

In particular, through the operator we extract whether *a certain topic of interest, regulated under a certain government, is also significantly reshaped by its direct successor in government.* Therefore, the output of such a rule is statistically significant patterns of discontinuity that allow us to monitor a possible important shift of regulation from one government to another regarding a certain topic. We highlight how such output is more high-level than looking for modifications and abrogations that a government made to the legislation implemented by its predecessor. In fact, the complexity of the legislative system hinders the result of such a query since these shifts could also be captured indirectly through third laws, which share the same (or similar) topic.

[2] Multiple approaches can be used to derive topics from textual documents, either semi-supervised ones [6] or state-of-the-art LLM-based topic extraction tools [34]

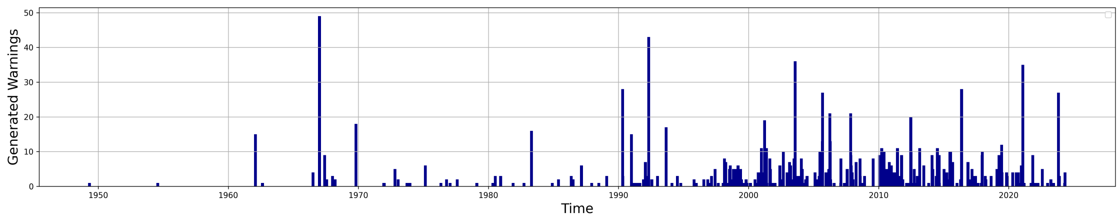

Figure 10 Warnings generated by our trigger-based reasoning system, aiming to monitor the abrogations of relevant laws. Each bar represents the number of warnings generated whenever a new law is published, i.e., the number of `UNDERMINES` relationships that are merged within the graph.

In addition, we also leveraged association rules to study other topological structures. For instance, we uncovered citation patterns from articles to laws. For instance, high-confidence rules reveal that articles citing “data protection” laws from the Economy and Communications departments consistently cite “privacy” legislation as well.

5 Implementation and Results

In this section, we execute the proposed triggers and association rules within the Italian legislation that we have already modelled in a property graph [16], aiming to demonstrate the efficacy and utility of such tools in a real-world scenario. In particular, we demonstrate how warnings generated by our triggers exhibit a significant correlation with ex-post interventions, i.e., correction measures that modify or add details to the law signalled by the alert system based on triggers. Then, we implement an AR in the Italian graph to analyze whether we can monitor and identify patterns in legislation related to procurement rules, which are under intense scrutiny in daily Italian news. We discover that in recent years, many governments have had a reshaping attitude toward such legislation, with profound changes and discontinuity among recent governments, which harm the stability of legislation, one of the drivers of economic growth [21].

5.1 Relevance of Warnings

As discussed in Section 4.1, we consider the emergence of a `UNDERMINES` edge, derived via reasoning with triggers, as a warning. We implemented the triggers in a Neo4j database, which we also used to store the graph of the Italian legislation. To implement triggers as reasoning rules, capable of inferring novel knowledge through deductions, as required by this use case, we made use of a publicly available Neo4j plugin[3] that, through the use of a trigger controller, manages the correct execution order of triggers that enable reasoning.

A full quantitative assessment is not feasible since there is no ground truth that allows the derivation of a confusion matrix. Thus, to test the significance of these warnings, we adopted a historical analysis strategy: we replicated the (ordered) insertion of each law within the graph to let the Neo4j triggers activate based on the legislative situation at each timestamp.

First, we computed the number of warnings generated for each law, which we depict in Figure 10. Then, we correlated these values with the number of ex-post interventions that the legislator had to make within the year after the law's publication to address or rectify a legislative void. We count interventions as the distinct number of references to the “undermining” law that occurs in the following 365 days after the official publication. Thus, for each law that activates the triggers,

[3] The implementation is available at [33]

Table 1 Sample of the warnings generated by a repealing law and the number of legislative interventions that have occurred, within one year after the publication of the repealing law, to third laws having as foundation a law abrogated by the "repealing" law.

Repealing Law	Warnings	Legislative Interventions
1949/264	1	4
1954/615	1	0
1967/18	49	20
1967/601	9	3
...	...	...

we get two values: the number of warnings generated and the number of interventions. In Table 1, we present a portion of the two values that we computed for laws that have been signalled as warnings.

Finally, we compute the Pearson correlation coefficient to measure the linear relationship between the two datasets. The idea is that a significant positive correlation coefficient implies that the higher the number of warnings generated, the higher the probability of necessary intervention to address the legislative void, indicating that warning counts are associated with the likelihood of subsequent intervention. Since the distributions of warnings and interventions are not normally distributed, we adopted a bootstrap approach [23, 40] to estimate the sampling distribution of the correlation coefficient and to test whether it is significantly different from zero. We observed a Pearson coefficient of 0.394 and a p-value of 0.02 (within a confidence interval of $[0.269, 0.506]$), indicating a significant positive relationship between the two series. This can be interpreted as a higher probability of intervention whenever more warnings are generated, i.e., a central law is repealed and requires more interventions.

While the positive correlation provides a quantitative foundation for the system's predictive power, we acknowledge that correlation alone does not account for all nuances of legislative complexity; for instance, it cannot establish direct causality or distinguish between the extension of different interventions. In this case, legal experts can be involved to evaluate and refine the reasoning process, ensuring the warnings effectively capture the most critical legislative voids and account for the material nature of the corrections.

5.2 Evolution of Procurement Regulation

As a representative example, we identified regulation regarding "procurement" as an interesting topic to analyze through association rules. The theme is under scrutiny by the general public for its long-standing episodes of corruption in the country, as also studied in dedicated economic research [22]. Through the running of our association rule tool for this topic, we aim to monitor the behaviour of governments in continuously modifying legislation in this area.

We implemented the `MINE GRAPH RULE` operator in Neo4j[4] as in previous work (see [10]). The output of the AR run for the procurement legislation is presented in Table 2. Considering the number of topics in the graph, setting the support and confidence thresholds to 0.2 effectively identifies strict rules. Out of the 69 governments of the Italian Republic, 16 have significantly reshaped the topics their predecessor had worked on. An alarming trend is that this "practice" has intensified in recent years. Notably, 15 out of the 16 governments identified began after 1990.

[4] The implementation is available at [27]

Table 2 Output of the `MINE GRAPH RULE` operator used to discover patterns of changes in procurement regulation. Results have been ordered by the temporal dimension, i.e., starting from the most recent pair of adjacent governments, and depict a scenario in which the government in the body column enacted relevant legislation about "procurement" which was profoundly reshaped by the (successor) government in the head column.

Head Government	**Body** Predecessor	**Support**	**Confidence**
I Meloni	I Draghi	0.57	0.80
II Conte	I Conte	0.29	0.66
I Conte	I Gentiloni	0.57	1.00
I Gentiloni	I Renzi	0.57	0.80
I Renzi	I Letta	0.28	1.00
I Letta	I Monti	0.28	0.66
I Monti	IV Berlusconi	0.57	0.80
IV Berlusconi	II Prodi	0.57	1
II Prodi	III Berlusconi	0.43	0.5
II Berlusconi	II Amato	0.29	1
II D'Alema	I D'Alema	0.29	0.33
I D'Alema	I Prodi	0.29	0.5
I Prodi	I Dini	0.29	0.5
I Ciampi	I Amato	0.29	1
VII Andreotti	VI Andreotti	0.43	0.75
III Moro	II Moro	0.43	1

Another concerning behavior highlighted by the extracted rules is that governments succeeding those of a different political alignment are more likely to repeal pieces of procurement legislation, suggesting a dismantling of the predecessors' work. For instance, one of the rules with the highest support and confidence values reveals that the fourth Berlusconi government (right-wing) reshaped procurement legislation previously modified or enacted by the second Prodi government (left-wing).

The utility of these extracted patterns extends beyond simple historical description, providing a quantitative metric for regulatory instability. High confidence values (e.g., 1.00 in several transitions) serve as a proxy for **legal uncertainty**, identifying periods where legislative continuity is entirely broken.

This transformation of the legislative graph into a set of predictable association rules provides a data-driven foundation for evaluating the quality and reliability of a country's institutional environment.

6 Adding the Regional Layer to the Graph

Legislative systems are complex not only because of how documents relate to each other, but also because they are built from many different sources. These sources contribute to a rich and layered structure of meanings and interpretations. So far, in the process of building the Italian legislative graph, our work has mainly focused on national laws issued by central institutions such as the Parliament and the Government. However, the Italian legal system is much more distributed, and there are other institutions, like regional governments, that produce legal documents. These additional sources must be taken into account to create a more complete and representative model of the legal landscape in Italy.

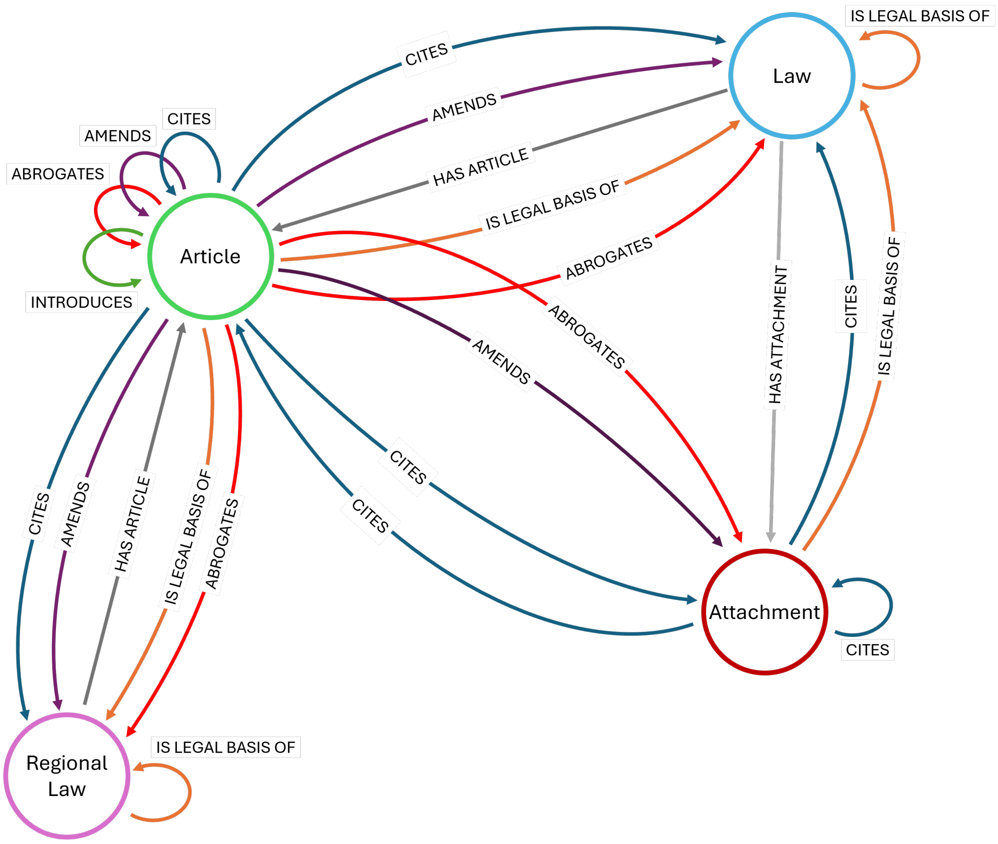

Figure 11 Graphical visualization of the updated Property Graph Schema including Regional Laws. The update schema included an additional node label `:RegionalLaw`, that has the same connections to `(:Article)` nodes as `(:Law)` nodes.

Therefore, as shown in Figure 11, we extended the Property Graph Schema. We introduced a new node label, `RegionalLaw`, which represents regional legislative documents. These nodes store metadata specific to regional laws but, like their national counterparts, they connect to `Article` nodes that contain the actual legal text.

In this updated model, `Article` nodes can reference both national and regional laws. They can cite other laws, serve as the legal basis for new laws, and participate in legal modifications such as amendments or abrogations. Importantly, these relationships are not limited to within the same jurisdiction: regional articles may refer to national laws, and vice versa, capturing the interconnected nature of Italy's multi-level legal system. Nevertheless, such documents are still laws, sharing all features of the national-level legislation, with respect to references and properties (e.g., type of law, publication date), and having specific features (e.g., region). Thus, the flexibility of the Property Graph Schema allows us to model such laws as an additional multi-labelled node, sharing the `Law` label with national nodes but also having the `RegionalLaw` label, which characterizes their nature.

Because each region follows different formats, handling all of them simultaneously would be highly complex and lead to inconsistencies in the resulting graph. To address this, we take an incremental approach: thanks to the flexibility of the property graph model, we start by incorporating regional data step by step. In this paper, we focused on refining the process for two regions, Lombardia and Liguria, and designed a general pipeline that will be implemented incrementally for all regions.

6.1 Lombardia

Regional laws from Lombardia are easily accessible thanks to a dedicated API provided by the regional government [31]. Each law is available in its most recent version, and, unfortunately, there is no consistent way to retrieve previous versions. The documents are provided in NIR-HTML format - which stands for *Norme in Rete* [7], an Italian standard used to structure legal documents.

References and citations to other laws can be extracted through specific HTML tags that contain the a unique code to the cited laws. To classify these relationships, we developed a set of heuristics that leverage the repetitive language commonly used in formal documents, allowing us to distinguish between abrogations, amendments, or general citations.

More than 2,500 law documents were processed and uploaded into the Italian Legislative Graph, including all laws promulgated by the regional government of Lombardia since 1971 when it started being active. For each law, a corresponding node was added, along with nodes for each of its articles, resulting in a total of 9,514 new `Article` nodes containing the text of the laws. In addition, over 9,000 edges were created. Of these, nearly 4,000 connect regional article nodes to articles in national laws, while the remaining edges represent intra-regional connections. Only a small fraction of these edges are labeled with something other than `CITES`, which may indicate the need for a more refined classification approach.

6.2 Liguria

Liguria's regional government does not provide an API to access its laws, but they are accessible through the official portal [29]. Each law is available in two formats: NIR-HTML, similar to the format used in Lombardia, and AKM-XML, also known as the Akoma Ntoso format. Akoma Ntoso (short for *Architecture for Knowledge-Oriented Management of African Normative Texts using Open Standards and Ontologies* [39]) is a widely adopted international standard for structuring legal documents in a machine-readable way.

However, NIR-HTML versions of Liguria's laws are only available for bulk download, which makes them impractical for building an end-to-end process that keeps the knowledge graph up to date as new laws are published. Instead, laws can be downloaded individually in the AKM-XML format, which requires a different process compared to Lombardia's laws. Even though the two process must be different, we aim to have a unified format for the processed data so that both regions can be uploaded and updated simultaneously daily.

Thanks to XML tags, we have direct references to citations to other laws. Using similar heuristics, we classified these references and label the corresponding edges in the knowledge graph.

The regional government of Liguria has published approximately 2,000 laws since 1971, and for the purpose of this work, we processed and uploaded them in the Italian Legislative Graph. In addition, more than 14,000 `Article` nodes have also been uploaded, containing the text of all the laws they refer to.

6.3 Pipeline for Regional Integration

The successful integration of Lombardia and Liguria highlights how regional governments publish legislative data in different formats. To extend the Italian Legislative Graph to cover all 20 regions, we have designed a generalized pipeline that processes this heterogeneity while ensuring the consistency of the final knowledge graph. We aim to implement this proposed pipeline in future works.

The pipeline is depicted in Figure 12. It is structured into four distinct steps:

- **Step 1. Region Law Extraction.** For each region, we have designed a specific extractor to fetch the missing data. For a region that needs to be added to the Italian legislation graph, it downloads the full corpus of laws. Instead, it only retrieves the laws that have been published the day before for those regions already in the graph.

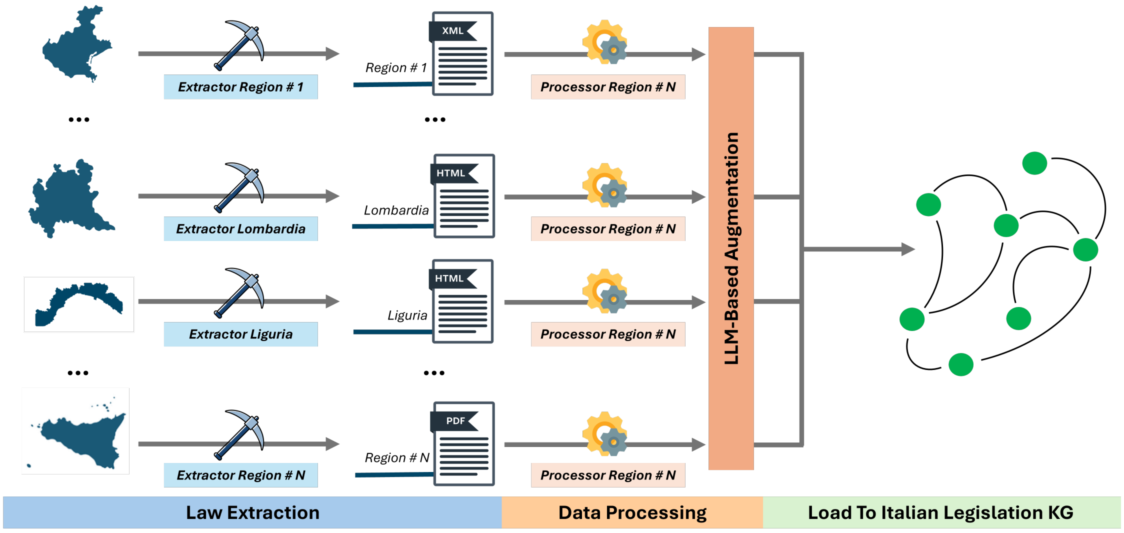

Figure 12 Proposed Pipeline for integrating Regional Laws into the Italian Legislation Graph.

- **Step 2. Region Data Processing.** Once the laws have been extracted, we process them to have a unified data structure. A specific processor is designed for the specific file format of the region. First, it splits the text of the law into granular Article nodes. It then extracts the metadata of the law, such as the regional law univocal code and its publication date. Lastly, through tailored heuristics and tags, it looks for citations to national laws, or to laws of the same or other regions.
- **Step 3. LLM-based Augmentation.** During this step, an enhancement layer using Large Language Models is applied to the data. LLMs are used to refine citation extraction, identifying references that traditional heuristic techniques fail to capture. Furthermore, this step assigns semantic topics to laws with LLMs, enriching the graph's metadata for advanced data queries.
- **Step 4. Data Loading.** The final step maps the processed and augmented data to the unified Property Graph Schema presented. It loads the regional laws into the graph database, merging with existing national and regional layers.

Edge Classification and Data Quality. The extraction pipeline adopts a hybrid approach to ensure data quality and accurate edge classification across different institutional layers. For national legislation, edge types are derived from official annotations provided by the original sources, ensuring high reliability in the primary dataset. However, regional governments (e.g., Lombardia and Liguria) often use heterogeneous formats and lack these standardized official annotations. Therefore, we used a BERT-based model fine-tuned on national legislation data and available in [18]. The model was fine-tuned from Italian Legal BERT [28], achieving 92% accuracy on the test classification set for national edges (5k random national edges, 4 unbalanced classes). We applied this model to the regional layer, whose citation recitals are similar to those of the national layer, although we expect slightly lower performance.

6.4 Graph Queries for Regional Legislative Knowledge Management

The introduction of regional laws adds a new layer to the legislative system and opens up new opportunities for managing legal knowledge using property graphs. Having all the data in a single graph instance makes it easy for applications to focus on individual regions, compare regions, and interact with regional and national laws.

```
MATCH (:Law)-[:HAS_ARTICLE]->(art:Article)
WHERE ('regioni' in art.topics or 'lombardia' in art.topics) and
NOT (EXISTS((art)<-[:CITES]-(:Article)<-[:HAS_ARTICLE]-(law:RegionalLaw)
     WHERE law.region = 'lombardia')
or EXISTS((art)-[:IS_LEGAL_BASIS_OF]->(law:RegionalLaw)
           WHERE law.region = 'lombardia'))
RETURN art
```

Figure 13 Query to detect national laws citing regional laws.

Table 3 Top three most recent results of the query in Figure 13.

Article Title	Article ID
2020|9#17	Cassa integrazione in deroga per Lombardia, Veneto ed Emilia Romagna
2002|81#1	Sospensione dei termini processuali e dei termini di prescrizione e decadenza per la Regione Lombardia
1998|6#22	Ulteriori interventi urgenti nei territori della Lombardia interessati dagli eventi idrogeologici del giugno 1997

For instance, a regional lawmaker might want to identify national laws that explicitly mention regional institutions in general, or their specific region in particular. This can help highlight areas where the region may need to take action, either by intervening or by passing new laws that align with or respond to national legislation.

In the query presented in Figure 13, we use the `topics` property of national `Article` nodes, which captures the main subjects addressed by the legal text. This allows the query to find all national legal documents that refer to regions in general or to a specific region (such as Lombardia in our example, though any region could be used). The query then filters out the documents that have already been addressed by the region – either by being cited or serving as the legal basis for other regional laws – highlighting those that may still require attention from the lawmaker.

Table 3 displays the three most recent articles that find no citation in the regional legislative articles. In all cases, such laws do not require a specific reference in regional legislation (as they concern exclusive national competencies). However, the query suggests that the regional government is up to date with national legislation and faces no urgent legislative backlogs.

Another example, reported in Figure 14, analyses how laws from different regions are interconnected. This would show how several regions rely on similar legal frameworks to govern areas that have not yet been addressed at the national level.

The query presented counts how many regions have directly cited a regional law that does not have a legal basis in any national law. It then returns only those regional laws with a count above a certain threshold – 1 in this example – which suggests that more than half of the Italian regions have cited that law. This would help national lawmakers identify regional laws that have gained widespread attention and may indicate a shared need for regulation at the national level. This example illustrates how modeling the legislative system as a property graph enables a deeper understanding of cross-regional legal patterns. While each region develops its own legal network, analyzing the connections between them can support more informed and coordinated decision-making at the national level.

For instance, Table 4 showcases the laws from Lombardy, citing other regions' legislative framework. The laws cover a wide array of regulatory domains, specifically transport regulations, animal usage in research, and wood logistics.

```
MATCH (law:RegionalLaw)-[:HAS_ARTICLE]->(art:Article)
          <-[:CITES]-(art2:Article)<-[:HAS_ARTICLE]-(law2:RegionalLaw)
WHERE law2.region <> law.region
WITH law, art, count(DISTINCT law2.region) AS countRegion
WHERE NOT EXISTS((:Law)-[:HAS_ARTICLE]->(:Article)-[:IS_LEGAL_BASIS_OF]->(law)) and
      countRegion >= 1
RETURN law, art
```

Figure 14 Query to detect regional laws that deserve the attention of national lawmakers.

Table 4 Three laws from Lombardia that have been cited by other regions.

Law ID	Law Title
2012|06	Disciplina del settore dei trasporti
2015|04	Norme in materia di protezione degli animali utilizzati ai fini di ricerca e sperimentazione e di promozione dei metodi alternativi
2016|24	Ratifica dell'accordo interregionale sul prelievo legnoso in ambito boschivo e sulla filiera legno

7 Conclusions and Perspectives

In this paper, we built upon recent efforts in modelling legislative systems within graph databases, describing the motivations and benefits of adopting Property Graphs as a powerful solution for comprehensive legislative knowledge management. We demonstrated how adopting a graph database and our schema can seamlessly handle the temporal aspects and evolution of legislative acts, delegating to queries the task of retrieving the correct information, such as computing the in-force text, instead of storing multiple versions of the same act.

Then, we looked at recent tools recently adopted and developed by the Property Graph community, i.e., triggers and association rules, to offer innovative and powerful applications that can be useful in conducting advanced legislative knowledge management. To this aim, we demonstrated how triggers can be used to build a simple but effective warning system that monitors whether legislative voids are created. Finally, we used association rules to discover significant patterns in the evolution of the Italian legislation referred to procurement, discovering an increasing trend of instability in such legislation. In future work, we will further explore the utility of these tools for knowledge discovery, refining their applications in the legislative domain with the help of experts and identifying new domains that could benefit from these advancements.

Finally, we began expanding the graph of Italian legislation to cover an additional layer of the legislative system. Specifically, we modified the graph schema to include laws issued by regional governments, and we integrated the regional laws of both Lombardia and Liguria, covering their legislative activity since 1971. This allowed us to incorporate historical laws and set up a system that keeps the graph up to date with new regional legislation. We then demonstrated how adding the regional layer enhances legislative knowledge management, using three Cypher query examples that explore connections within a single region, between different regions, and between regional and national laws.

Our work suggests that Property Graphs offer distinct advantages for legislative knowledge management compared to other approaches exploiting graphs, like RDF and Knowledge Graphs.

First, the ability to store rich properties directly on edges allows the model to naturally handle both the structural relationships and the document content, specifically enabling efficient law versioning without the complex logic or redundancy often required in RDF-based temporal modeling.

Second, the inherent flexibility of the Property Graph Schema facilitates the evolution of the data model, allowing for the integration of multi-level hierarchies, like the regional layer introduced in this study, which can be more challenging to expand in rigid structural formats.

Finally, while analytical techniques for RDF have existed for some time, this paper demonstrates how some recently developed tools designed for Property Graphs can effectively close this gap. Thanks to this, Property Graphs are a robust choice for modelling and handling legislative data.

Our main focus in the future will be to scale the integration pipeline to cover all twenty Italian regions. While the current implementation validates the approach for Lombardia and Liguria, expanding to the remaining regions requires tailoring extractors and processors for their specific publication formats. Futhermore, we will leverage the proposed LLM-based augmentation to enhance metadata and citation extraction, matching the quality of the data of national laws. With a fully populated graph, we also aim to design regional-level triggers and to extend association rule mining to better study interactions between national and regional layers.

Finally, we propose developing a user interface to make these graph-based tools accessible to non-technical stakeholders. Furthermore, we intend to integrate in the interface linguistic metrics, such as readability and syntactic complexity. Thanks to this, we aim to offer a tool designed to assess both the structural coherence and the textual quality of the legislation.

While this study focuses on the Italian legislative system, the underlying framework is designed for cross-jurisdictional portability.

First, since the model utilizes the universal OASIS Akoma Ntoso standard, it can be easily applied to other legislation. This has already been proven viable by applying the same schema to United States legislation [20] and to Switzerland legislation [15]. Then, the extractors and classifiers used to handle relationships would need to be adjusted on the specific legal constructs of the target jurisdiction.

Statement on Supplementary Materials and Additional Resources

The Italian Legislation Graph dataset is stored at [14], and the supporting query files are provided in the GitHub repository at [9]. We used version 5.26 of Neo4j.

References

1 Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, 1993. doi:10.1145/170035.170072.

2 Renzo Angles, Angela Bonifati, Stefania Dumbrava, George Fletcher, Alastair Green, Jan Hidders, Bei Li, Leonid Libkin, Victor Marsault, Wim Martens, Filip Murlak, Stefan Plantikow, Ognjen Savkovic, Michael Schmidt, Juan Sequeda, Slawek Staworko, Dominik Tomaszuk, Hannes Voigt, Domagoj Vrgoc, Mingxi Wu, and Dusan Zivkovic. Pg-schema: Schemas for property graphs. *Proc. ACM Manag. Data*, 1(2), June 2023. doi:10.1145/3589778.

3 Tara Athan, Harold Boley, Guido Governatori, Monica Palmirani, Adrian Paschke, and Adam Wyner. Oasis legalruleml. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law*, ICAIL '13, pages 3–12, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2514601.2514603.

4 Gioele Barabucci, Luca Cervone, Monica Palmirani, Silvio Peroni, and Fabio Vitali. Multi-layer markup and ontological structures in akoma ntoso. In *International Workshop on AI Approaches to the Complexity of Legal Systems*, pages 133–149, Beijing, China, 2009. Springer, Springer. doi:10.1007/978-3-642-16524-5_9.

5 Luigi Bellomarini, Anna Bernasconi, Stefano Ceri, Alessia Gagliardi, Davide Magnanimi, Davide Martinenghi, et al. Towards a standard for trig-

gers in property graphs. In *CEUR WORKSHOP PROCEEDINGS*, volume 3741, pages 70–79. CEUR-WS, 2024. URL: https://ceur-ws.org/Vol-3741/paper02.pdf.

6 Anna Bernasconi, Francesco Invernici, Stefano Ceri, et al. TETYS: Towards the next-generation open-source web topic explorer. In *CEUR WORKSHOP PROCEEDINGS*, volume 3692, pages 26–33. CEUR-WS, 2024. URL: https://ceur-ws.org/Vol-3692/paper4.pdf.

7 C Biagioli, E Francesconi, P Spinosa, and M Taddei. The NIR project: Standards and tools for legislative drafting and legal document web publication. In *Proceedings of ICAIL workshop on e-government: modelling norms and concepts as key issues*, pages 69–78, 2003.

8 A. Bonifati, D. Braga, A. Campi, and S. Ceri. Active xquery. In *Proceedings 18th International Conference on Data Engineering*, pages 403–412, 2002. doi:10.1109/ICDE.2002.994754.

9 Francesco Cambria. Cypher queries for legislative property graphs. swhId: swh:1:dir:c175195a74440b097ec11f0fb49fdc67cf616930 (visited on 2026-08-12). URL: https://github.com/FrancescoCambria/Cypher-Queries-TGDK26.git.

10 Francesco Cambria, Francesco Invernici, Anna Bernasconi, and Stefano Ceri. Mine graph rule: A new gql operator for mining association rules in property graph databases. *The VLDB Journal*, 34(4):1–21, 2025.

11 Stefano Ceri, Anna Bernasconi, Alessia Gagliardi, Davide Martinenghi, Luigi Bellomarini, and Davide Magnanimi. Pg-triggers: Triggers for property graphs. In *Companion of the 2024 International Conference on Management of Data*, SIGMOD/PODS '24, pages 373–385, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3626246.3653386.

12 Stefano Ceri and Jennifer Widom. Deriving incremental production rules for deductive data. *Information Systems*, 19(6):467–490, 1994. doi:10.1016/0306-4379(94)90002-7.

13 S Wesley Changchien and Tzu-Chuen Lu. Mining association rules procedure to support online recommendation by customers and products fragmentation. *Expert systems with applications*, 20(4):325–335, 2001. doi:10.1016/S0957-4174(01)00017-3.

14 Andrea Colombo. Italian legislation graph. https://zenodo.org/records/13798158.

15 Andrea Colombo. Switzerland legislation graph. https://zenodo.org/records/17615989.

16 Andrea Colombo. Knowledge graph of the italian legislation, 2024. doi:10.5281/ZENODO.13798158.

17 Andrea Colombo. Leveraging knowledge graphs and llms to support and monitor legislative systems. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, pages 5443–5446, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3627673.3680268.

18 Andrea Colombo. Bert-edges (revision 7fadf37), 2026. doi:10.57967/hf/8346.

19 Andrea Colombo, Anna Bernasconi, and Stefano Ceri. An llm-assisted etl pipeline to build a high-quality knowledge graph of the italian legislation. *Information Processing and Management*, 62(4):104082, 2025. doi:10.1016/j.ipm.2025.104082.

20 Andrea Colombo and Francesco Cambria. Llm-assisted construction of the united states legislative graph. *Proceedings of the VLDB Endowment. ISSN*, 2150:8097, 2025.

21 Kenneth W Dam. *The law-growth nexus: The rule of law and economic development.* Rowman & Littlefield, 2007.

22 Francesco Decarolis, Raymond Fisman, Paolo Pinotti, and Silvia Vannutelli. Rules, discretion, and corruption in procurement: Evidence from italian government contracting. Technical report, National Bureau of Economic Research, 2020.

23 Bradley Efron. *Bootstrap methods: Another look at the Jackknife*, pages 569–593. Springer New York, New York, NY, 1992. doi:10.1007/978-1-4612-4380-9_41.

24 Kapali P. Eswaran. Aspects of a trigger subsystem in an integrated database system. In *Proceedings of the 2nd International Conference on Software Engineering*, ICSE '76, pages 243–250, Washington, DC, USA, 1976. IEEE Computer Society Press. URL: http://dl.acm.org/citation.cfm?id=807683.

25 Amélie Gheerbrant, Leonid Libkin, Liat Peterfreund, and Alexandra Rogova. GQL and SQL/PGQ: Theoretical models and expressive power. *arXiv preprint arXiv:2409.01102*, 2024. doi:10.48550/arXiv.2409.01102.

26 Rinke Hoekstra, Joost Breuker, Marcello Di Bello, Alexander Boer, et al. The LKIF core ontology of basic legal concepts. *CEUR Workshop Proceedings*, 321:43–63, 2007. 2nd Workshop on Legal Ontologies and Artificial Intelligence Techniques, LOAIT 2007 ; Conference date: 04-06-2007 Through 04-06-2007. URL: https://ceur-ws.org/Vol-321/paper3.pdf.

27 Francesco Invernici and Francesco Cambria. Neo4j Mine Graph Rule Operator. https://github.com/FrInve/mine_graph_rule, 2024.

28 Daniele Licari and Giovanni Comandè. ITALIAN-LEGAL-BERT: A Pre-trained Transformer Language Model for Italian Law. In Danai Symeonidou, Ran Yu, Davide Ceolin, María Poveda-Villalón, Davide Audrito, Luigi Di Caro, Francesca Grasso, Roberto Nai, Emilio Sulis, Fajar J. Ekaputra, Oliver Kutz, and Nicolas Troquard, editors, *Companion Proceedings of the 23rd International Conference on Knowledge Engineering and Knowledge Management*, volume 3256 of *CEUR Workshop Proceedings*, Bozen-Bolzano, Italy, September 2022. CEUR. ISSN: 1613-0073. URL: https://ceur-ws.org/Vol-3256/#km4law3.

29 Regione Liguria. Banca dati della regione liguria. https://lrv.regione.liguria.it/liguriass_prod/.

30 Wolf Linder. Swiss legislation in the era of globalisation: A quantitative assessment of federal legislation (1983-2007). *Swiss Political Science Review*, 20(2):223–231, 2014. doi:10.1111/spsr.12100.

31 Regione Lombardia. CRL – Leggi regionali della lombardia. https://www.dati.lombardia.it/government/

CRL-Leggi-Regionali-della-Lombardia/abjw-hhay/data_preview.

32 C Lupo, Fabio Vitali, E Francesconi, Monica Palmirani, R Winkels, E de Maat, A Boer, P Mascellani, et al. General XML format (s) for legal sources. *Technical report, IST-2004-027655 ESTRELLA European project for Standardised Transparent Representations in order to Extend Legal Accessibility*, 2007.

33 Davide Magnanimi. Neo4j APOC light-weight reasoner. https://github.com/davidemagnanimi/apoc, 2023.

34 Reza Yousefi Maragheh, Chenhao Fang, Charan Chand Irugu, Parth Parikh, Jason Cho, Jianpeng Xu, Saranyan Sukumar, Malay Patel, Evren Korpeoglu, Sushant Kumar, et al. LLM-take: Theme-aware keyword extraction using large language models. In *2023 IEEE International Conference on Big Data (BigData)*, pages 4318–4324. IEEE, 2023.

35 Jim Melton and Alan R Simon. *SQL: 1999: Understanding relational language components.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

36 OASIS. Akoma Ntoso Version 1.0 becomes an OASIS Standard, 2018. https://www.oasis-open.org/news/announcements/akoma-ntoso-version-1-0-becomes-an-oasis-standard/.

37 Francisco de Oliveira and Jose Maria Parente de Oliveira. A RDF-based graph to representing and searching parts of legal documents. *Artificial Intelligence and Law*, 32(3):667–695, 2024. doi:10.1007/S10506-023-09364-9.

38 Timur Osadchiy, Ivan Poliakov, Patrick Olivier, Maisie Rowland, and Emma Foster. Recommender system based on pairwise association rules. *Expert Systems with Applications*, 115:535–542, 2019. doi:10.1016/J.ESWA.2018.07.077.

39 Monica Palmirani and Fabio Vitali. Akoma-ntoso for legal documents. *Legislative XML for the Semantic Web: Principles, Models, Standards for Document Management*, pages 75–100, 2011.

40 Jeffrey L Rasmussen. Estimating correlation coefficients: Bootstrap and parametric approaches. *Psychological Bulletin*, 101(1):136, 1987.

41 Stefan Schmitz, Markus Schluetter, and Ulrich Epple. Automation of automation — definition, components and challenges. In *2009 IEEE Conference on Emerging Technologies and Factory Automation*, pages 1–7, 2009. doi:10.1109/ETFA.2009.5347197.

42 Huayou Si, Jiayong Zhou, Zhihui Chen, Jian Wan, Neal N Xiong, Wei Zhang, and Athanasios V Vasilakos. Association rules mining among interests and applications for users on social networks. *IEEE Access*, 7:116014–116026, 2019. doi:10.1109/ACCESS.2019.2925819.

43 Francesco Sovrano, Monica Palmirani, Fabio Vitali, et al. Legal knowledge extraction for knowledge graph based question-answering. *Frontiers in Artificial Intelligence and Applications*, 334:143–153, 2020. doi:10.3233/FAIA200858.

44 Ramakrishnan Srikant and Rakesh Agrawal. Mining generalized association rules. *Future generation computer systems*, 13(2-3):161–180, 1997. doi:10.1016/S0167-739X(97)00019-8.

45 Meera Tandan, Yogesh Acharya, Suresh Pokharel, and Mohan Timilsina. Discovering symptom patterns of COVID-19 patients using association rule mining. *Computers in biology and medicine*, 131:104249, 2021. doi:10.1016/J.COMPBIOMED.2021.104249.

46 Fabio Vitali, Monica Palmirani, et al. Akoma ntoso: Flexibility and customization to meet different legal traditions. *BALISAGE SERIES ON MARKUP TECHNOLOGIES*, 24:1–9, 2019.

47 Jennifer Widom and Stefano Ceri. *Active database systems: Triggers and rules for advanced database processing.* Morgan Kaufmann, 1995.