Information Processing and Management 62 (2025) 104082

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

An LLM-assisted ETL pipeline to build a high-quality knowledge graph of the Italian legislation

Andrea Colombo *, Anna Bernasconi, Stefano Ceri

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via Giuseppe Ponzio, 34, Milan, 20133, Italy

ARTICLE INFO

Keywords:
Law
Knowledge graph
Property graph
Large language models
Data quality

ABSTRACT

The increasing complexity of legislative systems, characterized by an ever-growing number of laws and their interdependencies, has highlighted the utility of Knowledge Graphs (KGs) as an effective data model for organizing such information, compared to traditional methods, often based on relational models, which struggle to efficiently represent interlinked data, such as references within laws, hindering efficient knowledge discovery.

A paradigm shift in modeling legislative data is already ongoing with the adoption of common international standards, predominantly XML-based, such as Akoma Ntoso (AKN) and the Legal Knowledge Interchange Format, which aim to capture fundamental aspects of laws shared across different legislations and simplify the task of creating Knowledge Graphs through the use of XML tags and identifiers. However, to enable advanced analysis and data discovery within these KGs, it is necessary to carefully check, complement, and enrich KG nodes and edges with properties, either metadata or additional derived knowledge, that enhance the quality and utility of the model, for instance, by leveraging the capabilities of state-of-the-art Large Language Models.

In this paper, we present an ETL pipeline for modeling and querying the Italian legislation in a Knowledge Graph, by adopting the property graph model and the AKN standard implemented in the Italian system. The property graph model offers a good compromise between knowledge representation and the possibility of performing graph analytics, which we consider essential for enabling advanced pattern detection. Then, we enhance the KG with valuable properties by employing carefully fine-tuned open-source LLMs, i.e., BERT and Mistral-7B models, which enrich and augment the quality of the KG, allowing in-depth analysis of legislative data.

1. Introduction

The adoption of emerging databases and knowledge representation technologies, such as Knowledge Graphs, has recently raised the attention of many communities looking for accessible and efficient approaches to represent complex knowledge. Among them, the computer law community has been very active in proposing Knowledge Graph solutions for representing complex domains, such as the legislative one, with laws interconnected through citations (Anelli et al., 2023; Angelidis, Chalkidis, Nikolaou, Soursos, & Koubarakis, 2018; Rodríguez-Doncel, Navas-Loro, Montiel-Ponsoda, & Casanovas, 2018).

When dealing with legislative data, one of the main challenges is the textual nature of laws, which, therefore, contains unstructured information. While LLM applications for direct text-to-KG construction offer a promising solution to this issue, their output is too inaccurate for obtaining a high-quality representation of textual data that ensures correctness in the schema (Dong,

* Corresponding author.
E-mail addresses: andrea1.colombo@polimi.it (A. Colombo), anna.bernasconi@polimi.it (A. Bernasconi), stefano.ceri@polimi.it (S. Ceri).

https://doi.org/10.1016/j.ipm.2025.104082
Received 12 September 2024; Received in revised form 27 December 2024; Accepted 27 January 2025
Available online 10 February 2025

2023; Mihindukulasooriya, Tiwari, Enguix, & Lata, 2023). To tackle this, the computer law community has devoted many efforts to proposing appropriate international standards to represent, within the same schema, laws enacted in different legislative systems. Most of previous work have utilized the eXtensible Markup Language (XML) format, a semi-structured data model which has been naturally used for modeling in the computer law community to represent textual data such as laws (Lupo et al., 2007).

XML tags can be easily mapped into reliable Knowledge Graphs that store laws and their articles as nodes, connected by citation edges (Sana & Suganthi, 2017). Relevant XML-based proposals include the Legal Knowledge Interchange Format (LKIF) (Hoekstra, Breuker, Marcello, & Boer, 2007), LegalRuleML (Athan et al., 2013) and Akoma Ntoso (Barabucci, Cervone, Palmirani, Peroni, & Vitali, 2009). The latter was recently officially adopted by many international and national bodies as a common standard (Vitali, Palmirani, et al., 2019); among these, it has been adopted by the Italian legislator (Palmirani, 2021), making it possible to implement solid pipelines based on this standard, including the derivation of a reliable Knowledge Graph of the Italian legislation.

While the availability of an XML standard simplifies the task of creating a Knowledge Graph for legislative data, its practical usefulness depends on (i) the choice of which graph model and schema is adopted and (ii) its richness in terms of nodes, edges, and especially properties, which allow users to perform advanced analysis over the KG. Regarding the former, based on our work with policy experts and researchers at the Istituto Einaudi per l'Economia e la Finanza (EIEF, 2024), we noticed the need for a computable and flexible representation of legislative knowledge. For the latter, we believe that LLMs could play a pivotal role in assisting an ETL pipeline by complementing and enriching the KG with additional graph objects, especially when specifically fine-tuned for information extraction tasks.

To this aim, we directed our attention to the advancements in graph databases and especially to the standardization of the Graph Query Language (GQL) (Deutsch et al., 2022) - the Property Graph query language; based on this, we propose the first Property Graph schema for modeling legislation data. In our schema, we leverage the flexibility and the advantages of the GQL to express in a compact and intuitive form Hogan et al. (2021) the complexity of legislative data, e.g., by utilizing advanced data structures as properties of nodes and edges, which allow us to seamlessly capture the temporal dimension (the natural evolution of laws naturally through time), one of the most critical features of this domain.

We implemented the schema and derived the Knowledge Graph of the Italian legislation, stored in Neo4j (the most popular property graph database Guia, Soares, & Bernardino, 2017; Solid IT consulting, 2024). To achieve that, we created an end-to-end ETL pipeline that, starting from the laws published in the XML format Akoma Ntoso through Normattiva (Istituto Poligrafico e Zecca dello Stato, 2024) – the official endpoint for the Italian legislation – applies a set of transformations that map the XML tags into graph objects, achieving a consistent, interlinked representation of the domain. Then, we integrated additional information that can be easily accessed from official legislative endpoints and applied error detection and correction steps by leveraging the graph structure. Finally, we employed LLMs to enhance the graph by complementing missing information or deriving additional properties, following the line of LLM-augmented Knowledge Graph approaches, which aim to leverage LLM capabilities for the task of graph completion and construction (Pan et al., 2024). To this aim, we adopted a combined few-shot and fine-tuning strategy to improve the results' quality of LLMs and allow the use of lighter models, which would benefit the pipeline's future sustainability. In particular, we used a BERT-based model to classify laws according to its *domain* and two Mistral-7B models to complement missing *titles* and to assign *topics* to laws, articles, and attachments. Few-shot learning allows us to enhance the performance of LLMs in the extraction of structured information in the form of relations or properties, as discussed in Wadhwa, Amir, and Wallace (2023) and Xu, Zhu, Wang, and Zhang (2023), while, by adopting a fine-tuning strategy, we can develop specialized but light LLMs that can effectively handle the extraction tasks.

Our ETL pipeline guarantees a regular update of the *property graph of the Italian national legislation*, by running a dedicated job on a daily basis that processes newly published laws on the official Gazette and integrates them within the graph ecosystem. To demonstrate the usefulness and potentialities of our KG, we explore its main features using graph queries that leverage its richness and the advanced data structures that are unlocked by the property graph model and by the functional LLM-derived properties. To this end, we compute metrics and statistics that characterize the Italian legislation, including patterns of legislative activity and trends in lawmaking, and uncover government-specific patterns by leveraging the domains and topics (e.g., to characterize the areas of government intervention). We are inspired by typical manually computed metrics used by independent legislation supervisor bodies for annual reporting purposes (Osservatorio sulla legislazione della Camera dei Deputati, 2023). We then discuss the quality of the final results by analyzing the KG under multiple dimensions (Wang et al., 2021; Xue & Zou, 2023); specifically, we analyze the accuracy of the KG – through an ad-hoc comparison exercise with the updated textual laws – the consistency – by highlighting how we handle contradictions in the data – the completeness – by illustrating the quantitative improvements at each step of our pipeline – the timeliness – by analyzing its updating performances – and finally the trustworthiness (of the data sources) and the interoperability – in terms of re-applicability of the same pipeline to other legislative systems.

The contributions of this paper can be summarized as follows:

- We propose the first Property Graph (PG) schema for modeling legislative data, capable of capturing the main complexities of the domain through the use of GQL-compliant data structures in a practical and compact schema.
- We implement an end-to-end ETL pipeline for creating a Knowledge Graph of the Italian legislation based on a solid XML-to-graph mapping technique. To this aim, we leverage the international Akoma Ntoso standard implemented by the Italian legislator and integrate the KG with additional legislative data. Then, we use the Graph Query Language paradigm to detect errors and identify inconsistencies in graph patterns.
- We craft a minimal suite of (light) fine-tuned LLMs that allow us to complement and enrich the KG by using, when possible, the graph itself for training and following the software Sustainability and Open Source guiding principles (Kukreja, Kumar, Purohit, Dasgupta, & Guha, 2024; Raiaan et al., 2024; Wu et al., 2022).

- We explore and discuss the quality of the resulting Knowledge Graph by analyzing its main features and evaluating it across multiple dimensions.

2. Related work

The use of graph databases and Knowledge Graph technologies in the computer law community has garnered significant attention in recent years. The Semantic Web community, in particular, has made important advancements by developing ontologies and the RDF (Resource Description Framework) paradigm to represent legal information into knowledge graphs, based on the RDF triple paradigm (Anelli et al., 2023). These works are vital for linking knowledge bases by offering unique identifiers across multiple domains. However, they are limited to using the edge-labeled graph data model, of which RDF graphs are a specific type (Angles et al., 2017). RDF operates on triples—comprising a subject, predicate, and object—that serve as statements describing relationships between the subject and object. These RDF graphs can be queried using SPARQL (Pérez, Arenas, & Gutierrez, 2009), the semantic query language. However, the recent advent of an international standard query language for property graphs (ISO, 2024) throws an opportunity for innovation in legislative systems.

An in-depth discussion of the pros and cons of adopting RDF or property graphs (PGs) is out-of-scope in this work. Here, by discussing related works, we recall the main differences between the models, hinting at the domain-specific advantages of adopting the property graphs when modeling legislative data. Property graphs model the data as a mixed, i.e., partially directed multigraph (Deutsch et al., 2022). Both nodes and edges can be labeled and present – possibly multiple – properties (that is, they are associated with property/value pairs). As laws can be naturally seen as nodes in a KG, the property graph model allows assigning specific features directly to the node (e.g., a list of topics regulated by the law). Additionally, as each legislation traditionally has some features that make it unique, a flexible attribute representation that assigns such specificities to the node is preferable (see discussion in Das, Srinivasan, Perry, Chong, & Banerjee, 2014). In RDF, attaching additional contextual information to individual triples is less trivial, possibly requiring *reification*, a technique that allows you to make statements about statements but that hinders querying performance, storage efficiency, and usability (Orlandi, Graux, & O'Sullivan, 2021). The property graph data model also allows a more natural expression of paths and graph patterns, as GQL facilitates the expression of path structures by imposing user-friendly syntactic restrictions (Francis et al., 2023). In RDF, instead, the tabular result format of SPARQL limits the natural expression of graph patterns (Libkin, Martens, & Vrgoč, 2016; Seaborne, 2013), making it harder to express paths of complex structures. For instance, in the legislative domain, the combination of paths and attributes is crucial for detecting patterns in the data, such as inconsistencies, which can be easily identified via graph traversal queries. Finally, an important feature for an ever-increasing domain as legislative systems is also performance, with PG being very suitable for a rapid relationship traversal (Ciglan, Averbuch, & Hluchy, 2012), while storage in RDF triples - especially when combined with reification - harms query performances (Robinson, Webber, & Eifrem, 2015).

Preliminary graph prototypes for modeling legislative systems have been implemented in Greece (Angelidis et al., 2018), and Spain (Rodríguez-Doncel et al., 2018) and in Italy (Anelli et al., 2023). In such models – all based on RDF – law nodes are linked via relationships such as "*amends*", "*derives from*", "*cites*". Nevertheless, each of such prototypes has limitations. First, the granularity is at the level of laws, and the articles are not considered as nodes of the graph. Some of them, such as the Spanish one (Rodríguez-Doncel et al., 2018), strongly rely on NLP and on named entity recognition techniques to build the KG, which might result in something of low quality since the task of correctly identifying a law with plain AI-based technique is a complex task, possibly generating omissions and inconsistencies (de Maat, Winkels, & van Engers, 2006; Sadeghian et al., 2018). The Italian prototype (Anelli et al., 2023) dedicates its efforts to developing tools for supporting the navigation of the Italian legislative system, as also highlighted by its proposed main use-case applications, which are mostly oriented to specific searches of laws and connections and to produce graphical visualizations (Crotti Junior et al., 2020; Curtotti & McCreath, 2012; Curtotti, McCreath, & Sridharan, 2013; Oliveira & Oliveira, 2023). Note also that none of the mentioned exercises leverage an international standard – such as the XML-based AKN standard – that would allow higher quality through the use of the XML tags supporting the mapping to the graph. Other works have leveraged XML files to build a legislative knowledge graph, for instance, by using mapping languages that, however, become country-specific (Crotti Junior, Orlandi, O'Sullivan, Dirschl, & Reul, 2019).

The development of pipelines that aim to extract and organize structured information from legislative documents has been a challenging task since the spread of digitalized versions of textual law documents. For instance, in Purpura and Hillard (2006), authors develop a classifier of the US congressional legislation that detects the topic of primary importance for a bill. However, their approach requires that an unranked list of topics is already available, thus limited to legislations that already provide such metadata. In Wulczyn et al. (2016), authors develop a table-parsing algorithm to extract budget allocations, by employing machine learning classifiers. However, as stated by the authors, much of the difficulty arose from the lack of a structured way of accessing the document components. More recently, applications of Natural Language Processing and LLMs, especially for constructing Knowledge Graph in the legal domain, are spreading (Sansone & Sperlí, 2022). One of these is the Lynx project (Moreno-Schneider et al., 2020), which combines multiple NL techniques on the legal corpus to construct a KG. However, recent works have tested LLMs' performance in the domain of a direct text-to-KG application, demonstrating that they still lack the flexibility to create high-quality KGs. At the same time, they can still be used as assistants to augment the factual accuracy of domain-specific KGs (Zhu et al., 2024).

Inspired by the development of the GQL standard and the international adoption of an XML standard for representing laws, in this work, we propose a novel paradigm for representing the complexities of legislative systems by leveraging state-of-the-art graph database technology and using LLMs as assistants to enrich the nodes and edges of our KG. Our choice offers a good trade-off between intuitiveness and flexibility in the data representation (Angles, 2018), combined with a more controlled use of LLMs.

Table 1
Primary building blocks of the Akoma Ntoso standard, which are used to represent laws across many legislation traditions.

	XML tag	Content
Metadata	FRBRthis	Unique identifier for the act, as per legislative rules
	docTitle	Law title.
	docType	Type of the law.
	docDate	Publication date of the law.
	authorialNote	Additional non-normative text containing relevant information about aspects of the law.
Law text	preface/header	Information about the title of the law, the (progressive) number that identifies the law, its date of introduction.
	preamble	Part of the text that states the legal basis and introduces the law.
	body	Main content of the law, it includes all basic units of the law.
	article/section/rule	Fundamental unit of the law, i.e., principal split of the body
	conclusions	Tag containing closing statements and signature of ministers.
	attachments	Textual or graphical documents that integrate the information of the body.
	heading	Title of the basic unit of the law body.
References	citations	Citations to the laws or articles that are the legal foundations of what is being enacted.
	activeMod	Block containing the amendments/repeals made to another document.
	textualMod	Tag indicating which type of modification will be applied.
	source	Inside a specific textualMod tag, it contains the article of the text where the modification is stated.
	destination	Similarly to the source tag, but referring to the law or article that is being modified.
	href	Identifier of the destination document or portion of the document for a citation.

3. Foundations and graph schema

Building on the recent adoption of common international XML-based law representation standards, we focus on the ingredients of the ETL pipeline employed to construct the Knowledge Graph of the Italian legislation. We first recall one of the most popular international standards that are being adopted in more and more countries. Then, we present the first schema for modeling legislative systems in Knowledge Graphs based on the property graph data model. In the following sections, we will apply such ingredients to the Italian legislation and enhance the Knowledge Graph quality by employing Large Language Models.

3.1. The XML Akoma Ntoso international standard

By adopting and leveraging an XML standard internationally adopted, the process of building, analyzing, and comparing legislative systems would be strongly accelerated. Among the XML standards, Akoma Ntoso stands out as one of the most promising ones since it has been officially adopted by numerous countries (Vitali et al., 2019). One of its key advantages is its ability to capture essential features standard to law documents across different systems, such as identifying the fundamental units of a law and supporting identifier tags for modeling references to other laws. The AKN standard's specifications have also been approved by the OASIS body (OASIS, 2018), signifying its high quality and interoperability across legislative systems. Among the most important institutions that have adopted AKN, we can find the European Parliament (European Union Publications Office, 2023), which is likely to encourage many EU member states to align their systems with this format.

In the USA, the Library of Congress has tried to convert the US Code into the AKN standard (Legix.Info, 2012). Akoma Ntoso has been officially adopted and implemented in Italy, with all its laws published in this format in its official portal *Normattiva*,[1] the UK,[2] Switzerland[3] and also by international institutions like the United Nations (UN System Chief Executives Board for Coordination, 2017) and the FAO (Palmirani, 2018). Therefore, while this work focuses on Italian legislation, the approach to building the Knowledge Graph will directly apply to other countries that have implemented and published laws in AKN.

AKN Main Building Blocks. Table 1 details the main AKN building blocks that we will consider in the next sections to develop our ETL pipeline and to build the Knowledge Graph of the Italian Legislation (see Section 4). The AKN tags are designed to capture slightly different aspects of multiple legislative – democratic – traditions. For instance, the *preamble* always captures the *formulas* used to state the "legal basis", i.e., other laws that are essential to provide legal foundations to the new law, and to describe the "enacting sentences", i.e., linguistic expressions that are regular for a given tradition and are used to introduce the text of the law. The AKN standard also defines many tags that can be used in parts of the *body* (chapter, section, article, rule, etc.), denoting the basic units of a legislative system. Such tag depends on the specific legislative tradition, e.g., *article* and *rule* are the same object for different legislations. Since we focus on Italian legislation, we will use the *article* tag to refer to the fundamental law unit. For instance, in the case of the American tradition, the law unit is a "section" of the law. Special attention is required for law and article citations. In fact, multiple types of citations exist and each is captured in a dedicated XML tag, depending on the citation

[1] https://www.normattiva.it/.
[2] https://www.legislation.gov.uk/.
[3] Since May 30th, 2022, all new publications are in AKN (https://www.fedlex.admin.ch/eli).

type. The AKN standard dedicates specific blocks to modifications, i.e., amendments or abrogations that change the content of other laws, and to preamble citations, i.e., references to other laws or articles that are the legal basis of the law. Other citations might appear throughout the text inside a generic *href* tag, which cannot be classified as modifications or as legal basis. Finally, in many legislative traditions, we witness the presence of *attachments*, i.e., additional documents in a textual or graphical form, for instance, tables, which do not appear in the *body* of the law for any practical or other reasons. For instance, an international agreement approved by the related law is always provided as an attachment. Such objects are captured inside dedicated XML tags, i.e., *attachement* docs. In the next section, we detail how each tag is leveraged to build the graph objects.

3.2. Property graph schema

To express our schema, we consider Cypher (Neo4J, 2024), a declarative query language for property graphs (Angles et al., 2017) which is very close to the recently standardized Graph Query Language (GQL) (ISO, 2024). Cypher is supported by Neo4j, one of the most popular graph database management systems (Francis et al., 2018), that we will adopt throughout this work. The property graph data model comprises nodes that can have labels and multiple attributes (that we will refer to as properties), as well as directed relationships that might also be labeled and have their attributes. In Fig. 1, we depict the proposed property graph schema.

3.2.1. Description of the graph database schema

This section thoroughly discusses the graph schema and the motivation behind the modeling choices.

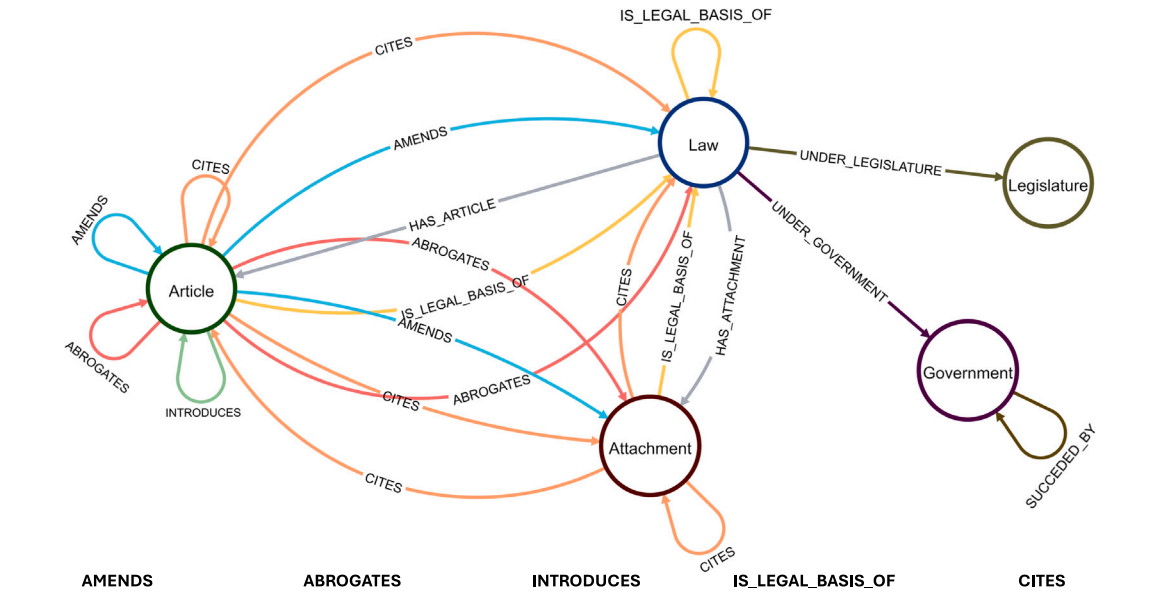
Law Nodes. Each enacted law is modeled as a node in the graph. Law nodes are identified by a string-based key adopted in each legislation. For instance, in the EU, the European Legislation Identifier (ELI) (European Union Publications Office, 2024) is used to identify unique legislative acts. In addition, we derive all relevant metadata and assign them as properties of the law; these include the title, the type of the law, the publication date, and the entry-into-force date. To improve usability, we also include among the node properties the number of articles and attachments; although a query could easily derive this information, we opt for its insertion as a property on the node, as it could be of immediate interest to users. Finally, we add the *domain* property (describing the ministries or departments involved in the new law) and the *topic* property (describing the specific topics that the law is addressing). These properties are modeled into *lists*, which can have multiple values (a feature supported by property graph data models).

Article Nodes. Each law comprises one or more *articles*, modeled as an additional schema node, connected by the `HAS_ARTICLE` relationship. The article is indeed the basic unit of the law, and it can always be identified through a progressive number, which is then concatenated to the law identifier to create an article ID. Articles have their properties: a title, a number, and the full text. To extract the full text of a law, it is sufficient to write a query that concatenates the text contained in (progressively) numbered articles (see Appendix A.2). In law nodes, topics are an additional property of practical utility; however, specific articles may treat distinct aspects of the law. For instance, a law titled "Provisions regarding the reorganization of the powers of the Departments" might dedicate its articles to each Department. Thus, each would be dedicated to a distinct topic.

Attachment Nodes. Possibly, a law can also comprise attachments or appendixes. These special documents specify practical aspects of the law or other documents (e.g., international agreements that must be implemented into the legislative national system). We model them as nodes, connected via the `HAS_ATTACHMENT` relationship to the parent law node. Attachments – by definition – do not contain any (direct) amendment or abrogation to other laws – which are always ruled by a law article. In other words, an article might rule modifications to other laws and indicate that such changes are detailed within an attachment. However, formally, the source of the modification remains the law article. Therefore, we consider them a distinct node type in our schema. Some attachments might be tables that specify additional information in tabular form. For instance, values of new tariffs for driving licenses or lists describing the reallocation of human resources between departments. Therefore, in addition to the same properties of article nodes, we add the *type* property, indicating the nature of the attachment's content.

Reference Edges. We model five types of possible reference edges that capture the interconnections between laws, articles, and attachments, namely, `IS_LEGAL_BASIS_OF`, `AMENDS`, `INTRODUCES`, `ABROGATES` and `CITES` edges. `IS_LEGAL_BASIS_OF` edges denote references in the introductory part of a document, which state its legal basis. Thus, the source node for such edges can be laws, articles, or attachments (contained in the preamble of the destination law). `AMENDS`, `INTRODUCES`, and `ABROGATES` edges are references that, respectively, substitute, add, or delete a portion or the full text of articles and attachments previously published. Per normative drafting rules (Karpen, 2008), attachments must feature content that cannot be phrased in a normative way (thus excluding modification rules). Consequently, these three kinds of edges always present an article node as a source. Instead, their destination could also be a law (e.g., when the law title is changed). Finally, `CITES` edges denote generic references to other laws that occur throughout the text to recall a relevant law, article, or attachment that might be important to cite for providing contextual information. For instance, a reference is used (and required) when providing a definition or specification of terms and objects used throughout the law, e.g., the list of harmful chemical substances.

Each of the described edges is assigned the *paragraph* property, a list indicating the destination paragraphs interested by the reference. If the reference points to the whole article or attachment, the *paragraph* property is null. `AMENDS`, `INTRODUCES`, and `ABROGATES` edges also present the *newText* property, which stores the text, as modified by the source node. `IS_LEGAL_BASIS_OF` and `CITES` edges are assigned the *weight* property, which counts the number of times that the same reference has appeared



AMENDS

Paragraph: LIST
NewText: STRING

ABROGATES

Paragraph: LIST
NewText: STRING

INTRODUCES

Paragraph: LIST
NewText: STRING

IS_LEGAL_BASIS_OF

Paragraph: LIST
Weight: INT

CITES

Paragraph: LIST
Weight: INT

Law

Property	DataType
ID	STRING
Title	STRING
TypeLaw	STRING
PublicationDate	DATE
InForceDate	DATE
NumArt	INT
NumAttach	INT
Domain	LIST
Topic	LIST

Article

Property	DataType
ID	STRING
Title	STRING
Number	INT
Text	STRING
Topic	LIST

Attachment

Property	DataType
ID	STRING
Title	STRING
Type	STRING
Text	STRING
Topic	LIST

Government

Property	DataType
Name	STRING
StartDate	DATE
EndDate	DATE

Legislature

Property	DataType
Name	STRING
StartDate	DATE
EndDate	DATE

Fig. 1. Property Graph schema visualization of nodes and directed edges, modeling a legislative system, such as the Italian one; the formal PG-Schema (along the definition of Angles et al. (2023)) is provided in Appendix A.1. On the bottom, we list edges and nodes with their properties; edges without specific properties are omitted for brevity.

throughout the preamble or the text.

Government and Legislature Nodes. Each law is enacted under distinct legislative landscapes, which we capture by adding the government and legislature nodes. Such nodes are naturally and univocally identified by their name, which is typically indicated with a progressive number, such as *Legislature I*, *Legislature II*, or *Berlusconi-I* and *Berlusconi-II* for governments, using the name of the Prime Minister in charge. Both graph objects share start and end date properties, which characterize the temporal evolution of legislature and governments. Note that, in most democratic countries, the start and end dates of legislature and governments do not coincide since governments usually stay officially in power even after a new parliament is elected, i.e., a new legislature begins; this motivates the absence of a direct edge between legislature and governments, while they are both connected to law nodes via the `UNDER_LEGISLATURE` and `UNDER_GOVERNMENT` edges. We model them as independent nodes in our schema. As we shall see in Section 5, this modeling choice proves helpful in deriving insights about the legislative landscape. For instance, by leveraging the `SUCCEDED_BY` relationship, recursive path queries allow the traversal of temporal patterns. In future iterations, more properties could be added to account for additional legislative landscape information, such as, for instance, the parliamentary composition in terms of political parties.

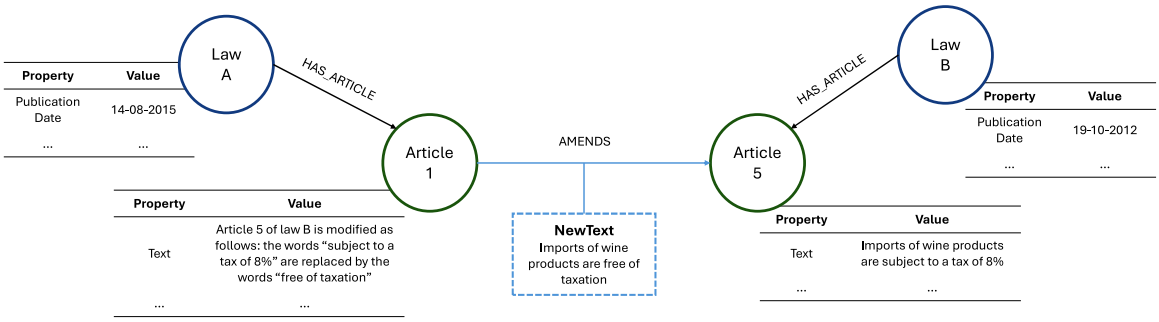


Fig. 2. Instantiated example illustrating how we capture the temporal evolution of laws. Here, Art. 5 of Law B, published in 2012, is amended by Art. 1 of Law A, published in 2015. The original text of Art. 5 is stored as a property of the node. Its new version, as modified by Art. 1 of Law A, is instead stored within the *NewText* property of the amend edge. As a consequence, the full text of Law B at a certain timestamp can be derived by a graph query (see Appendix A.2).

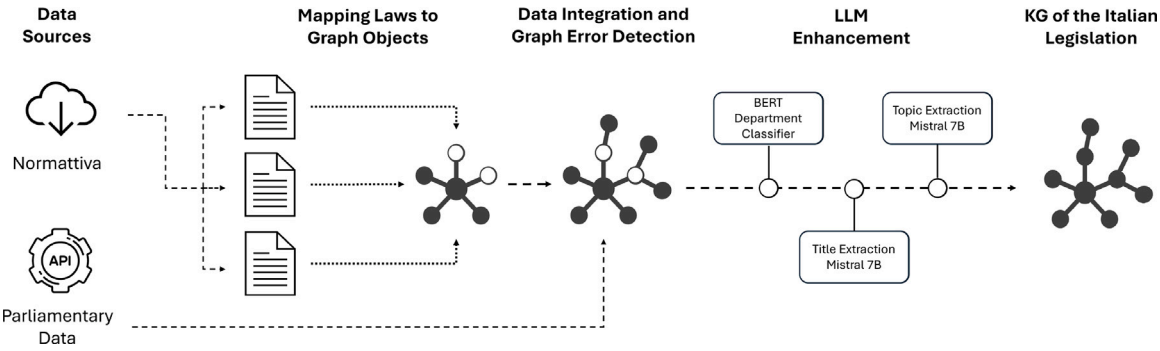


Fig. 3. ETL pipeline to build the Knowledge Graph of the Italian Legislation, adopting the property graph model. Normattiva is the primary data source for newly published laws, which are mapped to graph objects by leveraging the AKN international standard. The API endpoint of the lower chamber of the Italian Parliament allows us to integrate additional data into the KG. Then, we apply graph-based cleaning steps to correct errors detected through queries. Finally, a set of fine-tuned LLMs integrate the KG with additional features that complement and enhance the quality and richness of the database.

3.2.2. Capturing the temporal dimension

Legislative systems naturally evolve, with a continuous flow of new laws that modify or abrogate old ones. Any of such changes denote new *versions* of the laws, which capture the new text whenever a change occurs; this leads to an exponential proliferation of textual documents since small changes in the text require storing an additional file. We leverage our graph data model to overcome such limitations by storing modified articles as a property of the edges. This allows us to retrieve the desired version of the law through a query over the KG. Indeed, we only store the original version of the law within nodes; any change can be retrieved by navigating the graph to query the information about the law in a specific timestamp through edges properties. Fig. 2 illustrates how we use property graph features to track multiple textual versions of the same article. In Appendix A.2, we report practical queries that illustrate how we query the property graph for temporal-dependent features, such as deriving the version of the law or detecting the number of laws still in force at a given timestamp.

4. Building the knowledge graph of the Italian legislation

In this section, we build on the foundations presented in Section 3. We develop an Extract-Transform-Load (ETL) pipeline of the Italian legislative system. We leverage the advancements in official legislative data modeling (i.e., the AKN standard) and present a series of techniques to convert such documents and their content into graph data objects (Sana & Suganthi, 2017), i.e., the ones of the presented property graph schema. To this aim, we also integrate publicly available data sources and show how we combine the input data with a set of carefully fine-tuned large language models to obtain a comprehensive and high-quality representation of the legislative data, as we shall also illustrate. An overview of the ETL pipeline is presented in Fig. 3. The pipeline runs daily and updates our property graph database on Neo4j, publicly available (in a frozen version) in a Zenodo repository (Colombo, 2024b). The update of the graph relies on official data sources (i.e., government and parliamentary data), which thus ensure a regular and timely publication of novel data whenever they become available, i.e., a new law is published.

Sustainability and Reproducibility Requirements. We designed the ETL pipeline by prioritizing sustainability (specifically, efficiency) and reproducibility. To this aim, we adopted: (i) light-weight LLMs (which contribute to streamlined operations, reducing the overall computational burden), and (ii) open-source LLMs (which can be hosted freely and are replicable). In Section 4.5, we discuss the potential generalization of our pipeline to other legislative systems.

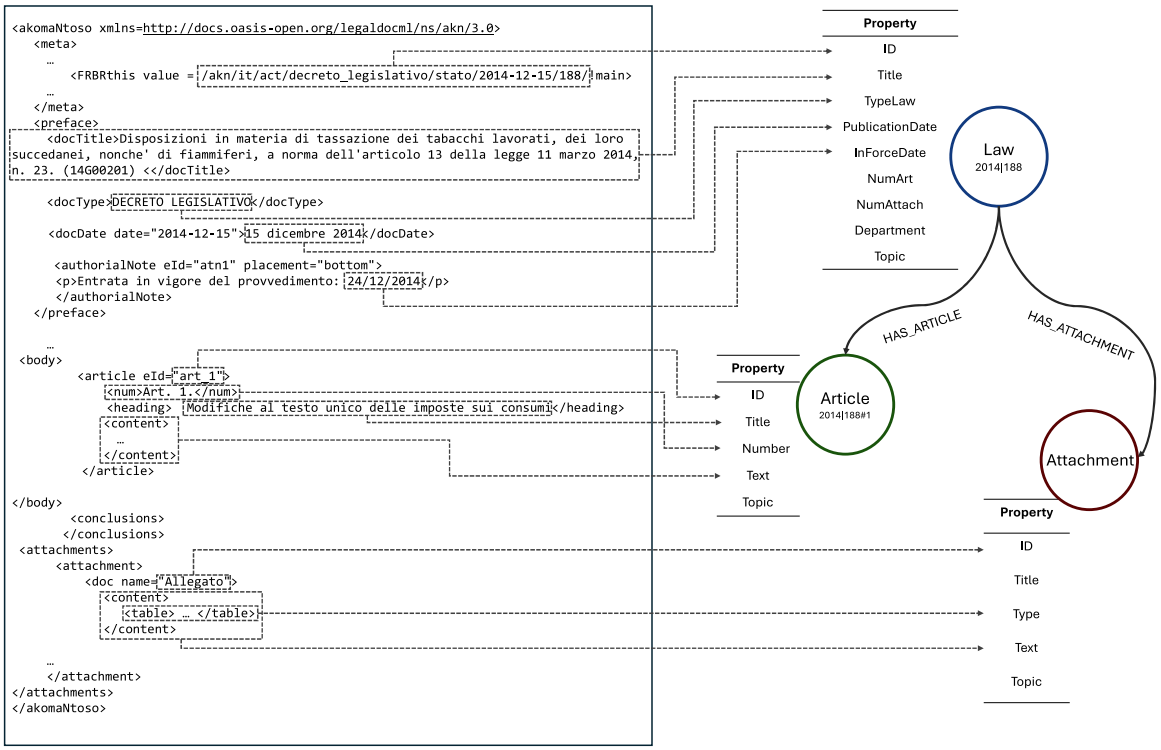


Fig. 4. Example of a mapping from AKN tags to graph nodes and their properties.

4.1. Data source of the Italian laws

The modern Italian legislative system dates back to the adoption of the republican Constitution in 1948, which serves as a cutoff point to exclude obsolete laws from the Kingdom period. However, we made exceptions for two significant laws, the Civil and Penal Codes, which remain in force despite extensive modifications. All laws are publicly accessible through the *Normattiva* portal, which utilizes the Akoma Ntoso standard. For the goal of building the Knowledge Graph adopting the schema presented in Section 3.2, we collected all laws published after the cut-off date in their original version, i.e., as they were published. Then, the pipeline automatically downloads newly enacted laws on a daily basis. While legislative systems typically differentiate between versions of a law, as discussed in Section 3.2.2, we only need to download new laws in their original form, as the proposed schema captures all subsequent changes.

Structure of Italian laws. As per official rules (Senato della Repubblica, 2001), Italian laws must follow a pre-defined generic structure. In the first part, after the title, a preamble indicates the legal foundations of the law, if available; then, the body of the law, containing articles which are split into *introductory articles* (with generic and principal rules of the law), *main articles* (with the detailed rules of what is being regulated), and *closing articles* (which contain information about the in-force terms of the act). Each article is divided into paragraphs, each ending with a line break. After the closing statements – containing signatures of responsible officials – tables, prospectuses, lists, etc., can be inserted in an annex to the legislative text.

4.2. Mapping AKN documents and tags to property graph objects

Laws in the Normattiva portal are available in the AKN standard (see Section 3.1); we present how we leverage the AKN tags of Table 1 to map its objects into a property graph, i.e., nodes, edges, and properties.

Schema Nodes. First, we derive the schema nodes, i.e., *laws*, *articles*, and *attachments* nodes, as presented in Section 3.2. Fig. 4 visually illustrates the mapping.

1. *Law Nodes*. Each law from Normattiva, i.e., each AKN document, represents a law node in the graph. For each law, the metadata captured within specific XML tags are used to derive and extract first node properties, as in Fig. 4. Specifically, we retrieve the title, the date of publication, and the type of act. By counting the presence of *article* and *attachment* tags available throughout the XML, we get the total count of articles and attachments. Then, we derive the in-force date property by searching in *auhtorialNote* tags for the tag '*Entrata in vigore del provvedimento*' (i.e., in-force date of the law), which contains a specific date.

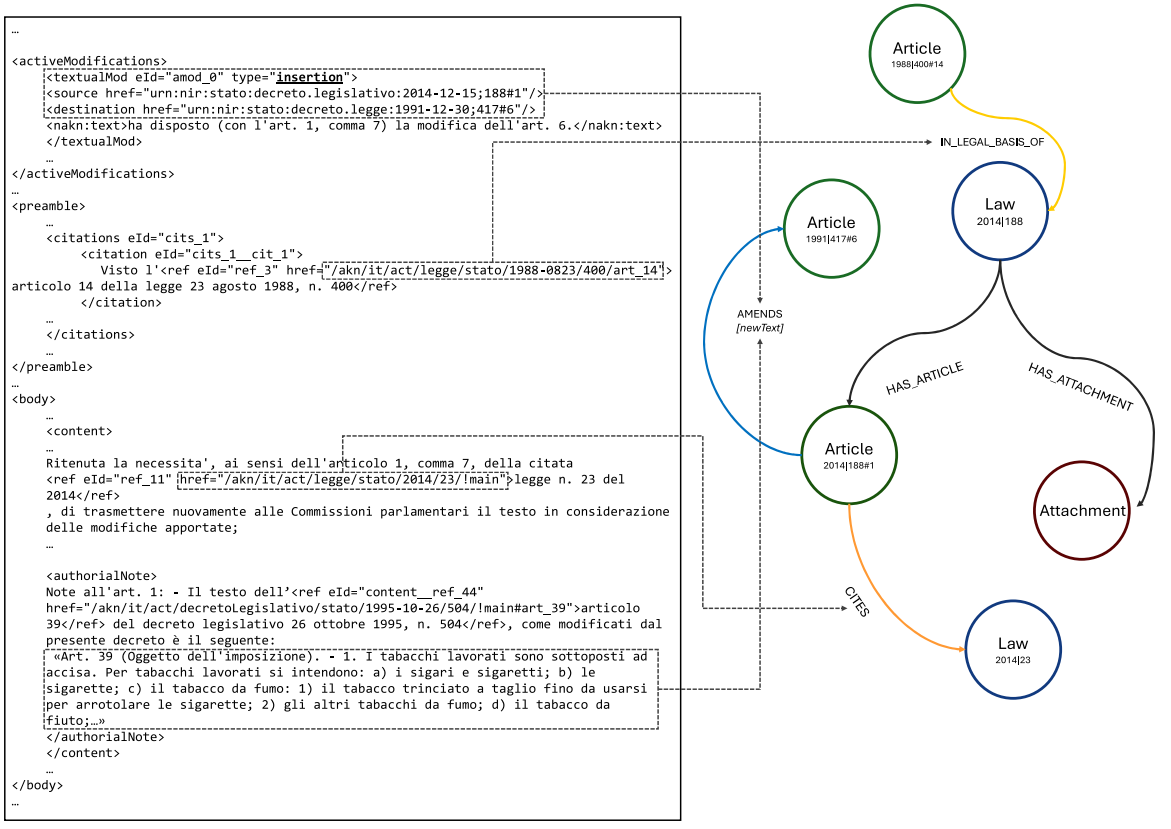


Fig. 5. Example of a mapping from AKN tags and their content to graph edges and their properties. Note that ABROGATES and INTRODUCES edges follow identical mappings as the AMENDS ones (whenever the corresponding *type* in AKN is respectively 'repeal' and 'introduction').

2. *Article Nodes*. In Italian legislation, each law is made of *articles*, the basic law unit that details different aspects of the same law. The articles have their own titles, called *epigraph*, and are captured within a specific *heading* tag. Thus, each *article* tag of the AKN file defines an article node in the graph, which is naturally connected to the parent law node that maps the AKN document in the graph. Each article has a *heading* tag, which is leveraged to retrieve the title property representing the epigraph, and a *num* tag, representing its progressive number within the law. We derive the ID of article nodes by concatenating the law node identifier with the article number. Such a construct is exploited in *href* tags throughout the text for citing other articles (allowing us to map also references, as we shall see in the following sections).
3. *Attachment Nodes*. Attachments are modeled as additional objects inside the *attachments* tags. For each attachment, the *doc* tag contains its name, which we use to compose the node ID, concatenated with the parent law id. For instance, if the attachment is a table, its identifier is derived by concatenation: *AKN_ID#Table 1*, where *Table 1* is the doc name. In addition, we derive the type of the attachment by looking at whether the *table* tag is used within the doc portion.

Law Reference Edges. Reference edges refer to citations to other documents present throughout the text. Citations might have different purposes and natures, such as substitutions of text, additions of new portions or words, or more generic references to recall certain aspects. We leverage the AKN standard to capture such distinction; the standard includes a pre-defined set of possible citations, together with dedicated XML tags that lawmakers must follow to comply with the international standard (i.e., substitution, insertion, split, join, renumbering, and repeal). In the following, we describe how we map AKN tags to KG objects (see Fig. 5 for visual support):

1. IS_LEGAL_BASIS_OF edges, representing the legal basis of a law. The AKN standard captures such references within the *citation* tags. We represent them as directed edges: their destination is always the law node whose preamble is being parsed; their source is another law, article, or an attachment of another law. The *weight* property counts how many times the same source–destination pair of a given type is mentioned throughout the text.
2. AMENDS edges. We derive modification edges by searching for the *textualMod* tags inside the *activeModifications* XML block. According to the standard, each *activeModification* tag represents a modification, i.e., an edge of our graph.
3. INTRODUCES edges. From amendments edges, we isolate the *textualMod* tags whose type is *insertion*-only and derive the introduction edges, which add additional text without modifying previous paragraphs.

4. ABROGATES edges. Similarly, the tags whose mode is *repeal* are instead modeled as abrogate edges, which deprecate part of the text. In case an article is completely repealed, the edge has no *paragraph* property, thus indicating a full abrogation.
5. CITES edges, i.e., other more generic citations to other legislation, that might be another law, an article, an attachment, or even specific paragraphs. Such citations complement the text by recalling other useful information and are gathered by detecting other *ref* blocks within the text of the law.

For all types of citations, we use a *list* data structure to capture the cases where multiple distinct paragraphs of the same article are cited. For instance, if there are two *amends* edges with the same source–destination pair -e.g., (Law A Art. 2)-[r:AMENDS]->(Law B Art. 1) – but referring to distinct destination paragraphs – e.g., paragraph 1 and 4 of Art. 1 Law B, respectively – we represent both within the same edge but with the paragraph property having two elements (e.g., r.paragraph = [1,4]).

The Role of AuthorialNote tags. The implementation of the AKN standard within the Italian system uses the *AuthorialNote* tags for adding annotation throughout the text. As an example, it is used to add the *inForceDate* among the metadata, since no specific AKN tag is defined for such information (see the mapping in Fig. 4). It is also used for inserting useful contextual information throughout the text of the law, such as the destination text as modified by an article of the law; for instance, in Fig. 5, Article 1 of Law 2014/188 is amending Article 6 of Law 1991/417 (see the *activeModifications* tag at the beginning of the XML document). The full new textual version of the latter can be retrieved by parsing the authorial note at the end of Article 1, which specifies the new version, comprising the modifications introduced by the new article. In other words, whenever the actual text of the law is ruling a substitution of some parts of another article, an authorial note – which is not part of the actual text of the law – indicates the new version of the destination article. In our pipeline, the new text is assigned as a property to the modification edges.

4.3. Data integration and graph-based error detection

To broaden the scope of our property graph, we first proceed with the integration of data that describes the contextual legislative landscape. Then, by employing graph queries, we detect errors and inconsistencies in the data, and whenever possible, we directly adopt a correction strategy aimed at improving the *consistency* of the KG. We also illustrate how we leverage the data model for signaling system inconsistencies to the lawmaker, i.e., errors that derive from incorrect legislative activity.

4.3.1. Legislature and government nodes

As per our schema, we integrate information about the government in charge and the legislature under which the law was published. Such information allows us, as we shall see in Section 5, to analyze features of the legislative systems on a higher level. In the case of the Italian legislation, we collect such data from the endpoint provided by the Italian Parliament (Camera dei Deputati, 2024), which provides up-to-date information about governments and parliamentary data. The edges connecting laws to governments and legislature nodes are derived by leveraging the temporal dimension to understand under which government and legislature a law was published.

4.3.2. Graph-based error detection

Although the source of AKN documents is of high quality (they are provided directly by the Official Gazette), a set of graph patterns can be run to check for inconsistencies in the data source. Such graph patterns can be easily implemented through our PG model in the form of Cypher queries. Note that errors' presence affects query results, for instance, when computing systemic statistics based on graph patterns; thus, their detection and reporting are paramount for achieving a high-quality representation of legislative data.

1. Auto-citation edges, i.e., reference edges with the same node as source and destination. While internal references are allowed, we exclude them from our KG as their nature differs from other citation edges. We identify them through a simple Cypher query (see below) and then remove them from the graph.

```
MATCH p=(l:Law)-[:HAS_ARTICLE]->(a:Article)-[:CITES|AMENDS|ABROGATES|INTRODUCES]->
    (a2:Article)<-[:HAS_ARTICLE]-(l:Law) RETURN p
UNION
MATCH p=(l:Law)<-[:IS_LEGAL_BASIS_OF]-(l:Law) RETURN p
UNION
MATCH p=(l:Law)<-[:IS_LEGAL_BASIS_OF]-(a:Article)<-[:HAS_ARTICLE]-(l:Law) RETURN p
```

   where *l* is the law that might be connected to itself through three distinct graph patterns, namely, references between its own articles, direct self-references in the preamble, and references to one of its articles in the preamble. This query allowed us to remove 90 edges, mostly generic CITES references (first pattern type of the query) and IS_LEGAL_BASIS_OF citations (second pattern type).
2. Incorrect edges source, i.e., abrogates, amends, or introduces references whose source article has been incorrectly inserted in the AKN document (see the *activeModification* tag in Fig. 5). We can derive such inconsistencies by running the following Cypher query[4]:

[4] This query is and can be run only in the update phase of the KG, and uniquely for novel nodes; thus, it is not affected by the *newtext* property.

```
MATCH p=(l1:Law)-[:HAS_ARTICLE]->(a1:Article)-[r:ABROGATES|AMENDS|INTRODUCES]
   ->(a2:Article)<-[:HAS_ARTICLE]-(l2:Law)
WHERE NOT a1.text CONTAINS toString(a2.number) // i.e., article number
 AND NOT a1.text CONTAINS split(l2.id,"|")[1] // i.e., law number
 AND NOT a1.text CONTAINS toString(l2.publicationDate.year)
RETURN p
```

detecting the graph pattern p where an article $a1$, source of an ABROGATES, AMENDS or INTRODUCES edge, does not contain any textual reference to $a2$ within its text (i.e., the article number, the law number, and the publication year of the law). This means that the source within the *activeModification* tag was incorrect. We identified about 4k edges affected by this issue and corrected them by searching among the other articles for the same pattern.

3. Re-classification of CITES edges into IS_LEGAL_BASIS_OF. We detected cases for which the preamble might be incorrectly inserted within the first article of the law. Consequently, citation edges within its text are identified by our pipeline as CITES edges, while they should be instead considered as IS_LEGAL_BASIS_OF edges. To fix that, we can run the following Cypher query:

```
MATCH (n:Article)-[r:CITES]->(s:Article)<-[:HAS_ARTICLE]-(l:Law)
WHERE toLower(n.text) CONTAINS "presidente della repubblica"
AND toLower(n.text) CONTAINS "decreta"
AND split(n.text,"decreta")[0] CONTAINS toString(l.publicationDate.year)
AND split(n.text,"decreta")[0] CONTAINS toString(s.number)
RETURN r
UNION
MATCH (n:Article)-[r:CITES]->(s:Law)
WHERE toLower(n.text) CONTAINS "presidente della repubblica"
AND toLower(n.text) CONTAINS "decreta"
AND split(n.text,"decreta")[0] CONTAINS toString(s.publicationDate.year)
AND split(n.text,"decreta")[0] CONTAINS split(s.id,"|")[1]
RETURN r
```

that leverages the preamble *formulas* (i.e., the presence of keywords that must be used to introduce the law) to detect edges r that are CITES and whose source articles n contain the preamble within their text. In particular, we use the presence of the *rituals*: "presidente della repubblica" (i.e., Republic's President) and "decreta" (i.e., enacts) as heuristics to identify and parse the articles whose text includes the preamble. While the former is the introductory ritual that characterizes the preamble of all laws of the Italian Republic, the latter is a ritual word that closes the preamble and introduces the text of the law. Through these heuristics, we derive the edges that have been incorrectly labeled as CITES ones, and we convert them into IS_LEGAL_BASIS_OF. Through this query, we identified 6273 edges that were incorrectly derived as CITES edges, and we converted them into IS_LEGAL_BASIS_OF ones. A total of 3255 distinct articles were affected by such inconsistency.

4. Articles used as the legal foundation of another law, but that were already abrogated at that timestamp. By tracking labeled edges in the property graph, we identify articles that have been cited after their abrogation, representing errors in the legislative system.

```
MATCH p=(l:Law)-[:HAS_ARTICLE]->(a:Article)<-[r:ABROGATES]-(a2:Article)
   <-[:HAS_ARTICLE]-(l2:Law)
MATCH (a)-[:IS_LEGAL_BASIS_OF]->(l3:Law)
WHERE r.paragraph IS NULL
AND l3.publicationDate > l2.publicationDate
RETURN l3.id as LawWithError, a.id as CitedAbrogatedArt
```

We detected 145 citation errors, relatively uniformly distributed across the years. The nature of such errors is different from previous inconsistencies: they originated from an incorrect drafting of the law throughout the lawmaking activity. Therefore, here, we do not apply corrections to the data but just observe how our data model can capture – and potentially report – such inconsistencies.

4.4. Enhancing the graph with large language models

While the nodes and edges of the property graph schema can be derived – and corrected – by implementing rule-based techniques, as described in previous sections, some relevant and useful properties are challenging to retrieve through plain heuristics or data integration techniques. This is the case of the *domain* law attribute, which specifies the ministries involved in the drafting of the law,

Table 2
Portion of the keyword-domain pairs we leverage to derive the domain from the ministry name. We identified a list of 16 possible domains: domestic affairs, institutions, agriculture, education, economy, communication, presidency, transportation, healthcare, foreign affairs, justice, labor, defense, public administration, arts and environment, sport, and tourism. The keywords allow us to univocally map a ministry to a domain.

Keyword in ministry name	Domain
aviation	transportation
health	healthcare
diplomacy	foreign affairs
pensions	labor
woods	agriculture
...	...

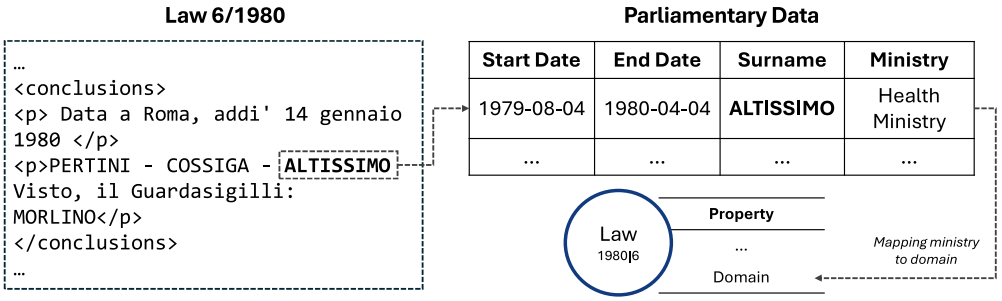


Fig. 6. An example deriving the domain of the law from the signatures in the conclusions of the law. To this aim, we integrate parliamentary data to get the ministry name and role and use the dictionary (on the top right corner) to derive the domain assigned to the corresponding law node. Note that the first two signatures are ignored since they always belong, respectively, to the Republic's President and to the Prime Minister.

a useful way to characterize laws for performing social and economic analysis (Giommoni, Morelli, & Paserman, 2022). Another feature of great utility is the *topics* of the law, which enables querying laws, articles, and attachments referring to the same content. Such information is also relevant for annual statistics, and their presence supports the automation of reporting activities (Osservatorio sulla legislazione della Camera dei Deputati, 2023). Furthermore, although the adopted XML-based publication standard remains extremely helpful in gathering the data needed to build the KG, we experienced deficiencies regarding the correct use of the standard and the published texts, especially for articles' titles. Note that titles are the primary element that summarizes the content of a law or an article, playing a crucial role for developing techniques that allow Retrieval Augmented Generation (RAG) of graphs or information retrieval pipelines on top of a legislative KG like ours. To tackle these deficiencies, we implemented some LLM-based steps that allow us to improve the *completeness* of the KG by integrating and enriching nodes with additional properties, capturing *domains*, *titles* and *topics*.

4.4.1. Ministry domain classifier

Each law can be classified according to the governmental departments – or ministries – that are involved in the drafting activity. The AKN standard does not specify any tag that captures such information. To derive it, we leverage the fact that any approved law must be signed by one or more ministries (who represent their ministry), a piece of information found in the *conclusions* of the law.

The first challenge derives from the fact that appointed ministers change over time, mainly due to the change of governments. In some cases, signatures in the conclusions are also decorated by the corresponding ministry name, such as *Franco, Ministro dell'Economia e delle Finanze*; however, in most laws, we found that the specification of the ministry name is missing as it is not a mandatory field but only an optional one. Here, we exploit the parliamentary data endpoint to get historical data regarding ministers and their departments, allowing us to link each surname to its ministry of reference (see Fig. 6). Out of 74k laws that are part of our database, about 65k laws required this linkage activity.

A second challenge is that ministry names also change over time; thus, they must be correctly grouped based on the actual domain. For instance, the treasury ministry has changed multiple names and was once also split into distinct ministries, namely *Ministero del Tesoro*, *Ministero dell'Economia*, and *Ministero delle Finanze*, which need to be traced back to the same domain, i.e., *economy*. Through the Italian Republic's history, we could identify 229 distinct ministry names (see Appendix A.3), which should be grouped by domain to perform temporal analysis. Based on this set, we manually crafted a set of keywords that uniquely link the ministry's name to a domain. Overall, we identified 16 domains and 107 keywords that can link the ministry name to a domain. Table 2 shows some examples of keyword-domain pairs that we use for detecting the domain based on keywords within the ministry name.

BERT-based Classifier. Although the combination of data integration and keyword-based approach allows us to derive the ministry for many laws, (i) some laws were incorrectly formatted and we could not derive the signatures (1902 cases) and (ii) aiming to

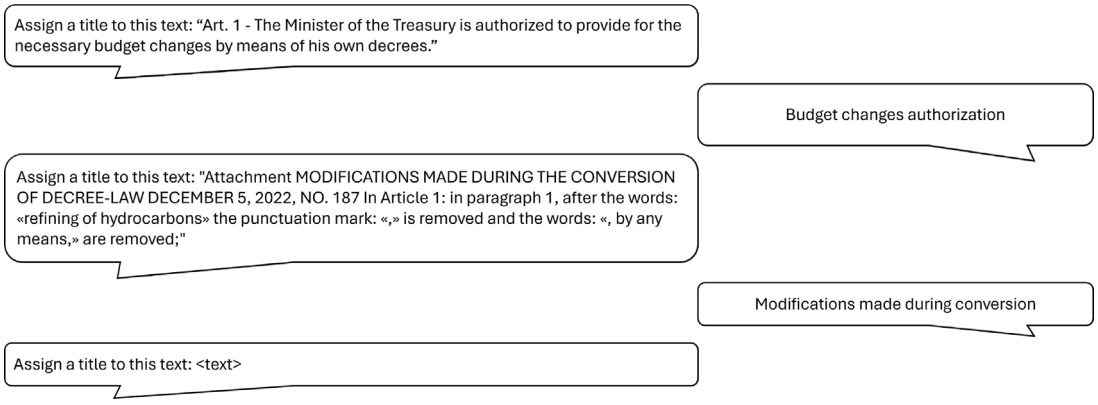


Fig. 7. Manually crafted title extraction examples provided to the LLM model for few-shot learning, i.e., instructing the model on the task it will perform via examples. The text under analysis is then passed within the third prompt.

build an end-to-end pipeline consistent over time, such deterministic approaches are not ideal, as they must be manually revisited every time a novel ministry name appears. To tackle this, we resort to an LLM-based solution, which provides more flexibility than the keyword-based vocabulary shown in Table 2. Indeed, it will not require future maintenance while also overcoming the issue of incorrect formatting of the AKN-based laws due to the absence of the ministry's signatures. By considering it as a multi-label classification problem, we created a dataset of 45k laws by connecting the law titles with the ministry/ministries derived from the deterministic approaches previously discussed. Then, we split the dataset into a 90–10 train and validation set. We considered a Bidirectional Encoder Representations from Transformers (BERT) model (Devlin, Chang, Lee, & Toutanova, 2018) that demonstrated state-of-the-art classification performances in various domains (Chen, Du, Allot, & Lu, 2022; Zahera, Elgendy, Jalota, Sherif, & Voorhees, 2019).

We fine-tuned the BERT model with an AdamW optimization (Loshchilov & Hutter, 2017) and used a sigmoid activation function to account for the multi-class nature of the problem, i.e., more domains for the same law. We trained the model for 5 epochs on a batch size of 32, with a learning rate of 2e−5. Training required around 40 min; epoch 4 was the best one we obtained, with an average training loss of 0.22 and an accuracy of 90%, which we consider quite high and acceptable given the additional challenge of dealing with the Italian language, which the model (trained mainly on English text Devlin et al., 2018) was not expert on. The details of fine-tuning steps, with training and validation loss at each step, can be found in Fig. A.1 in Appendix.

Although more powerful Large Language Models are available to date, given the overall good performances of the BERT fine-tuning, we decided to rely on this model. This choice is in agreement with our sustainability and reproducibility requirements, where we also take into account the ethical consequences of using very large models for simple tasks (Gunasekar et al., 2023; Ray, 2023) like multi-label classification. Our fine-tuned BERT model is available on Huggingface (Colombo, 2024a). Therefore, this model has been used to *complete* the KG by deriving the domains for all law nodes.

4.4.2. Article titles extraction

In our schema, we assign titles to all law-related nodes (i.e., laws, articles, and attachments). This information is useful for building information retrieval pipelines (e.g., RAG), which must identify relevant text based on textual input. While it would be possible to use the text of the article or the attachment, the pipelines would perform worse when presented with long texts (Wang, Huang, & Sheng, 2024).

While the law title is always available and captured by the metadata within the *docTitle* AKN tag, we experienced significant errors in the *heading* tag of articles: out of 318k articles, only 108k had a heading tag. For attachments, no title tag is even used. To tackle this, we implemented an LLM-based step to derive a title from custom text, i.e., the article's content. Again, here we could have applied state-of-the-art, very large, pre-trained Large Language Models, which are capable of dealing with both the Italian language and returning directly to the law title. Instead, consistent with the sustainability requirements for our pipeline, we opted for a smaller model, which we can fine-tune to achieve performances similar to larger ones.

Mistral-7B. In its *instruct* chat-based version, Mistral 7B is a 7-billion-parameter language model that achieves a good balance between accuracy and computational efficiency (Jiang et al., 2023). It is significantly smaller than larger models like GPT-4 or Llama3-70B, outperforming other comparable large language models. It is released under the Apache 2.0 license, allowing users an easy way of fine-tuning for specific tasks. Its smaller size, with overall good performances and open-source availability, made it a good choice for our pipeline.

We first experimented directly with the available pre-trained Mistral-7B model for the title extraction task, by providing the textual content of the articles and prompting with examples, performing few-shot learning, an approach where a model learns to generalize from a very small amount of training data that are directly provided as contextual input to the model (Parnami & Lee, 2022; Wang, Yao, Kwok, & Ni, 2020). Fig. 7 illustrates the examples provided for few-shot learning. Still, the model gave

Table 3
Differences between the pre-trained Mistral7B model and our fine-tuned version in the task of title extraction when provided the same system content, i.e., by performing the same few-shot learning. Examples refer to title extraction for articles that the model has not seen during training.

Text	Mistral-7B Pre-Trained	Mistral-7B Fine-tuned
Article 13. 1. This Protocol enters into force on the date agreed upon by the Parties through a subsequent exchange of notes. 2. The Protocol remains in force for 5 years. Unless one of the two Parties communicates, with at least six months' notice before the expiration, its intention not to renew...	Article 13	Final provisions and duration of the Protocol
Art. 7. 1. The regime established by the law of December 1, 1948, no. 1438, and subsequent amendments, limited to the product identified with no. 13 in Table A, attached to the law of December 27, 1975, no. 700, is redefined according to the provisions of this article, in accordance with Articles 30 and 32 of the cooperation agreement between the European Economic Community and the Republic of Slovenia...	Modifications to the regime of concessions for the product identified with no. 13 in Table A	Modifications to the regime of concessions for fuel in Slovenia
Art. 33. 1. Volunteer organizations may employ employees or utilize self-employed or other types of labor only to the extent necessary for their regular operation or to the extent required to qualify...	Article 33. volunteer organizations	Funding of volunteer organizations

unsatisfactory results since, after a manual inspection, the title was often not self-explanatory of the article's content or too generic (see examples in Table 3). In rare cases (1%), we also experienced answers in languages different from the target one, i.e., Italian.

Fine-tuning for the Title Extraction Task. To improve the results obtained from the LLM, we fine-tuned the Mistral-7B model. We built a large training dataset by gathering articles whose headings were available (i.e., the corresponding AKN tag was filled in correctly); this allowed us to gather a total of 108k high-quality title-text pairs, with the desired language (Italian) and referring to the task of interest (law article titles extraction). We adopted the Low-Rank Adaptation (LoRA) technique, which allows for quick adaptation of LLMs to specific tasks by keeping the original pre-trained weights frozen and only training newly introduced trainable parameters (Hu et al., 2021). Mistral 7B has been shown to perform slightly better in task specialization (Zhao et al., 2024), making it ideal for our pipeline.

Results. The model has been trained for 5 epochs with batch size of 4, 4-bit quantization using bitsandbytes and a LoRA rank of 64. We use the paged Adam optimizer, a learning rate of 0.004, and a cosine learning rate scheduler with a 0.03 warm-up fraction. We used an A100 GPU with 40 GB of memory, and the best model reported an evaluation loss of 1.003 (available on HuggingFace Colombo, 2024c). Training required around 9 h. Details about the evolution of training and validation losses are depicted in Fig. A.1 in Appendix.

The fine-tuned model is included in our pipeline as an additional component that enriches the article nodes with titles when unavailable, i.e., the compiled AKN documents fail to report titles. Since the nature of the task and the content are very similar and correlated, we also employed the same model to derive titles for attachments.

4.4.3. Topic extraction

While domains are useful as a high-level type of classification, their scope is still too broad compared to the large set of *topics* that laws and articles might regulate. Instead, topics are keywords that briefly capture the content of the text (of laws, articles, or attachments) and that might continuously change over time.

While the title already gives information about the content of the law, it does not help in performing structured queries — its content might frequently vary due to slight changes within the text. Deriving topics requires (i) identifying the characterizing keywords of a text and (ii) generalizing keywords to root/more frequently used – and strongly related – words (going beyond the specificity of the actual keyword). While the former can be achieved with NLP techniques or unsupervised systems, the latter can be achieved only by employing state-of-the-art large language models capable of capturing concepts semantics and going beyond plain keyword detection (Invernici, Bernasconi, & Ceri, 2024; Mu, Dong, Bontcheva, & Song, 2024; Wu, Gong, Shou, Liang, & Jiang, 2023).

Furthermore, LLMs can seamlessly adapt and account for emerging topics (e.g., artificial intelligence regulations), ensuring that novel trends are captured. For instance, consider two law titles that contain, respectively, the words *covid vaccines* and *SARS-CoV-2 virus*, which could be identified as keywords. An optimal common topic for both cases would be *covid-19*, allowing us to query both laws with the same more generic keyword.

Fine-tuning for Topic Extraction. Similarly to what was done in the title extraction task, we fine-tuned another Mistral-7B model to perform the task of topic extraction. Here the task is more challenging since we do not have a training set that can be deterministically derived from the AKN documents. To tackle this issue, we first resorted to LLMs to derive the topics from law titles. Big models (such as Mixtral-8×22B, GPT-4, or LLama-3 70B) are well-performing in the topic extraction from text (He, Huang, & Li, 2024), also when dealing with the Italian language. We considered a Mixtral-8×22B model, and we provided it with some examples for few-shot learning together with the title of the laws (see examples in Fig. 8). We created a dataset using the law titles, prompting the model with the following instructions: *Extract the topics from this title: <text>* and as system context: *You are an assistant who extracts topics from titles. Each topic must have a few words. Return only a concise list.*

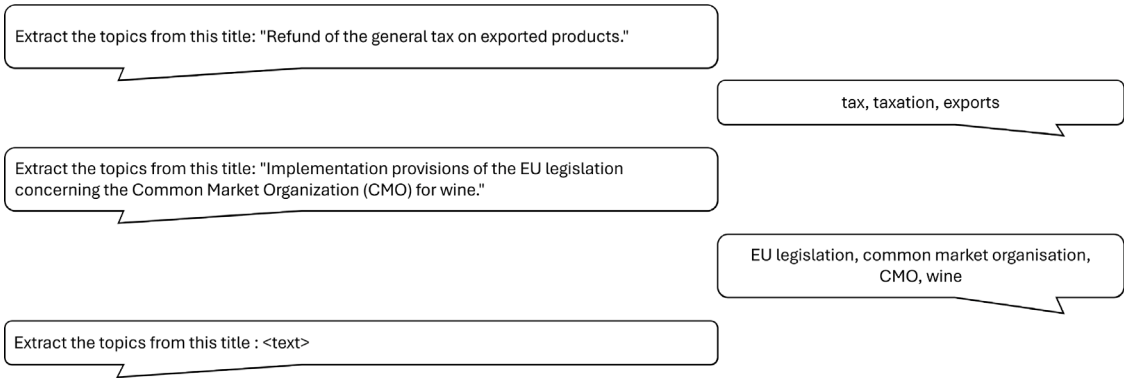


Fig. 8. Manually crafted topic extraction examples provided to the LLM for few-shot learning. The process is similar to the one used for the title extraction task (see Fig. 7).

Table 4
Differences between the topics extracted with the pre-trained Mixtral-8 × 22B and with the (fine-tuned) smaller Mistral-7B. While some topics are common, the pre-trained model shows lower generalization capabilities and some irrelevant and repeated topics, e.g., the second row.

Title	Mixtral-8 × 22B Pre-Trained	Mistral-7B Fine-tuned
Capital Increase Characteristics	Capital, increase, characteristics	Capital increase, corporate finance
Entry into Force	In-force date, activation, regulatory activation, regular activation, activation	In-force date
Increase in Annual Personal Contribution	Personal contribution, year, increase	Personal contribution, pension
Powers of Control and Search by Police Forces	Police, control, powers	Police, control

Although increasing the model size mitigated issues in the output, we experienced mixed quality, often extracting irrelevant topics, such as the law type/date or the number of the laws cited in the title. By randomly sampling a set of 4k law nodes, we observed that the most frequent topics were "regulation" (19% of the cases), "law" (10%), "ratification" (9%), and "decree" (7%). Besides "ratification", we noticed how such topics were insufficient to characterize specific aspects of a law. To tackle this, before performing fine-tuning to our smaller model, we analyzed the dataset of title-topic pairs, and we applied string-based heuristics to (i) reduce the number of irrelevant topics in the dataset and (ii) harmonize them, i.e., deriving the *root* such that we could account for distinct declination of the same word. Specifically, we dropped generic most-frequent keywords and used the law type/date node properties to drop topics related to this feature. Then, we *lemmatized* the words by employing spaCy (Honnibal, Montani, Landeghem, & Boyd, 2020), a multi-language state-of-the-art tool to reduce a word to its base or root form, known as a "lemma". This allowed us to improve the quality of the training set for a more meaningful topic extraction.

Training and Inference. We used the same configuration of the fine-tuning of the title extraction model and obtained an evaluation loss of 0.61 for the best model (available on HuggingFace Colombo, 2024d). Similarly to the title extraction model, training required around 9 h. Details about the evolution of training and validation losses are also shown in Fig. A.1 in Appendix.

Then, we used the fine-tuned model to derive topics both for articles and attachments. The model is regularly run to derive topics also for newly published laws. In Table 4, we show an excerpt of the results of topic extraction obtained with the Mixtral-8×22B pre-trained model and with Mistral-7B after fine-tuning it over some article titles, which are outside the training set.

4.5. Generalization to other legislative systems

While we implemented the full pipeline for the Italian legislative system, the components of our ETL pipeline can be replicated for other legislations with appropriate slight changes. The main ones depend on the publication approach adopted by each legislation. Many countries are currently developing APIs or other machine-friendly interfaces to improve data access. Table 5 reports an overview of the official data sources of legislation for six major countries, together with the availability of an API and a machine-readable publication standard that could be used for building a graph-based resource similar to the one described in this paper.

At the time of writing, the UK is in the experimental phase of developing its own API[5] that adopts the AKN international standard (as discussed in Section 3.1); mapping laws to graph objects of our pipeline would require only small adaptations, i.e., modifying the AKN tag conventions that account for the UK legislative tradition. In the US, an API for accessing legislation (US Library of Congress, 2024) was recently developed; the AKN standard has not been adopted, but laws are described using an XML-based format, this

[5] https://www.legislation.gov.uk/index.

Table 5
Overview of legislative data sources and availability across multiple countries.

Country	Data source	API availability	Publication format
United States	Library of Congress	Yes	National-specific XML
United Kingdom	legislation.gov.uk	Experimental phase	AKN
Germany	Bundesgesetzblatt	No	National-specific XML
France	Légifrance	Yes	PDF Only
Spain	Boletín Oficial del Estado	Yes	National-specific XML
Switzerland	Fedlex	Yes (SPARQL Endpoint)	AKN

Table 6
Statistics characterizing the law-related nodes (Laws, Articles, and Attachments) with the interconnections of reference and the "parthood" ones. Government and Legislature nodes are omitted, as they only link law nodes.

		Destination node		
	Graph node	**Law**	**Article**	**Attachment**
Source node	**Law**	61.989 Is Legal Basis of	318.286 Has Article	126.674 Has Attachment
	Article	44.954 Is Legal Basis of 7.009 Amends 532 Abrogates 78.256 Cites	70.990 Amends 5.123 Introduces 62.050 Abrogates 95.214 Cites	3.561 Amends 2.393 Abrogates 4.922 Cites
	Attachment	113 Is Legal Basis of 29.557 Cites	19.295 Cites	1.173 Cites

provides a basis for mapping laws and their content to our graph schema objects. The same is valid for Germany and Spain, which publish legislations in their own XML format; in Germany, the possibility of adopting the AKN standard is under investigation (Flatt, Langner, & Leps, 2022). In France, Légifrance offers access to all legal acts produced nationally. However, only PDFs are available, thus requiring a more challenging step of identifying the structure of each law before transforming the data into a graph.

Once the graph is derived, the graph-based error detection component of our pipeline (discussed in Section 4.3.2) does not require any adaptation due to the presence of a unifying abstract schema of the Knowledge Graph. The last step of the pipeline is the LLM-enhancement steps of the KG, which can be replicated in other systems by (i) training country-specific models on available data to integrate missing information or, in worst case scenario, (ii) adopting bigger LLMs that with few-shot learning can still perform the tasks of information extraction from text (Wadhwa et al., 2023; Xu et al., 2023), also considering many different languages (OpenAI, 2024),without the need of fine-tuning, although at the cost of harming the sustainability of the pipeline.

Finally, a notable system worth mentioning is the Swiss one, since (i) it has recently adopted the AKN standard and (ii) it publishes laws, between the others, in Italian, given the multilingual nature of the country. Thus, if an API is made available, the pipeline would be applicable without little or no adaptations, including the LLMs that we fine-tuned for the Italian language.

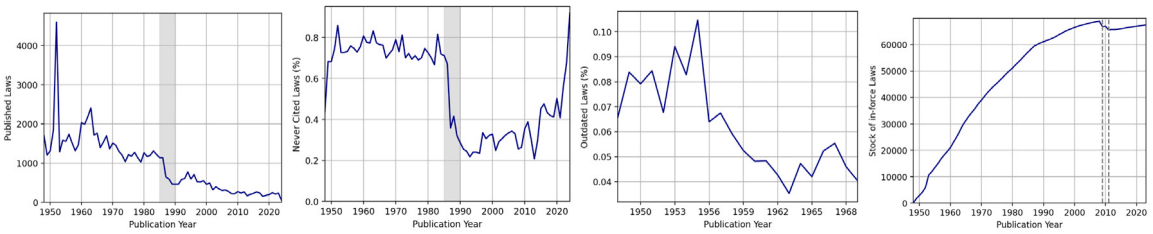
5. Knowledge graph exploration via graph queries

In this section, we explore the KG resulting from our ETL pipeline by illustrating the main features of the Italian legislation. To this aim, we also propose distinct types of graph queries facilitated by our data model. They respond to typical statistics that are (manually) computed by statistics offices for annual reporting purposes (e.g., Osservatorio sulla legislazione della Camera dei Deputati, 2023) or for developing interactive applications to monitor the legislative system (Colombo, 2024). By leveraging the produced property graph, we show how such activities can be supported by the data model and by the ETL pipeline that we implemented. In Table 6, we illustrate the main Property Graph dimensions. We modeled over 500k nodes and over 1 million edges, including references and edges expressing parthood.

Temporal Features of the Italian Legislation. During the 80 s and 90 s, we can observe a radical shift in how laws are drafted: while the annual number of laws has decreased significantly, the length of each law has increased, with more articles and attachments per law. Although analyzing the reasons behind this change is outside the scope of our paper, it is essential to take this trend into account when conducting queries, as it can significantly affect the results. For example, a straightforward query to identify the governments that produced the most laws might be skewed by this trend. Finally, special consideration should be given to the so-called *Decreti Semplificazione*. These decrees include numerous repealing rules intended to scrap and clean the legislative landscape of obsolete acts. When performing queries involving abrogate edges, users may want to filter out such laws, including laws 2008/112, 2010/66, and 2010/212. Additionally, a set of queries can be designed to gain general insights into the temporal evolution of the Italian legislative system by filtering and aggregating attributes based on criteria such as year, legislature, or government.

For instance, consider the following queries, whose Cypher statements are in Appendix A.7 and results plotted in Fig. 9:

Q1 *Laws published per year of publication*, meaning a count of laws based on their publication year.
Q2 *Laws never cited after publication*, referring to laws that have not been referenced in any preamble, nor have they received any amendments, abrogations, introductions, or other citations after their date of publication.



(a) Published Laws (Q1) (b) Laws Never Cited (Q2) (c) Outdated Laws (Q3) (d) Stock of in-force Laws (Q4)

Fig. 9. Panel (a) plots the results of Q1, which detects the number of published laws per year. The initial peak that can be observed is caused by the shift from the monarchy to the Republic, which required many changes in the legislation. Panel (b) shows the result of Q2 (the fraction of laws that have received no citation after the publication), which also highlights how, reasonably, many recently published laws are not cited yet. In panel (c), we plot Q3; here, we considered the cut-off date of 1970. Then, we used D = 1992 – the start of the "second republic" in Italian politics to compute the fraction of outdated laws. Panel (d) shows the results of Q4; here, we highlight the drops corresponding to simplification decrees (in 2008 and 2010, respectively).

	Conte I (461 Days)	Conte II (527 Days)	Draghi I (616 Days)	Meloni I (608 Days)
Agriculture	0.03	0.06	0.04	0.09
Arts and Environment	0.15	0.1	0.15	0.08
Defense	0.04	0.07	0.09	0.15
Economy	0.43	0.49	0.44	0.5
Foreign affairs	0.26	0.31	0.27	0.21
Justice	0.26	0.28	0.24	0.25
Domestic Affairs	0.14	0.12	0.13	0.27
Institutions	0.12	0.18	0	0.18
Education	0.08	0.07	0.06	0.11
Labor	0.11	0.05	0.1	0.14
Presidency	0.18	0.11	0.16	0.13
Public Administration	0.13	0.08	0.11	0.12
Healthcare	0.07	0.16	0.15	0.15
Sport and Tourism	0	0.03	0.02	0.08
Transportation	0.11	0.1	0.11	0.15

Fig. 10. Heatmap with the relative number of laws signed by ministries of the last four governments from 2018 to 2024 (in parenthesis, their duration in days), as computed by the graph query Q5, illustrating the domain focus of each government. Since laws can be multi-domain, i.e., signed by more than one ministry, the sum over columns is higher than the unit. Note that some values might be zero when a government does not appoint a minister for a specific domain.

Q3 *Outdated laws*, defined as laws that have stopped being cited after a certain point in time. To identify these, we first select laws published before a specific cut-off date. Then, by choosing a subsequent date D, such as one marking a significant political event, we extract the set of laws cited by any legislation published after D. This helps us identify laws not cited after D.

Q4 *Stock of in-force laws*, referring to the total number of laws in effect on a given date. This involves identifying which laws have not been abrogated by that time. In the context of Italian law, the official data source Normattiva allows users to view whether a law is in force or repealed at a specific point in time. However, this requires retrieving all laws with the desired date selected. Alternatively, by leveraging abrogate edges in the knowledge graph, we can determine which laws have been abrogated — when all of their articles have been abrogated or when the entire law has been directly repealed.

Exploring Law Domains and Topics. By leveraging the additional node properties of domains and topics, it is also possible to compute, by running queries on the graph, metrics that characterize the legislative system, such as the ones (otherwise manually) calculated for the annual reports of the legislative activity in Osservatorio sulla legislazione della Camera dei Deputati (2023). Such reports present statistics to the general public, summarizing tendencies and features that characterize certain legislatures. Similar and more advanced statistics and visualizations can be achieved by querying our graph. For instance, let us consider the proposed queries, whose Cypher statements are available in Appendix A.7:

Fig. 11. Word cloud visualization of the results of query Q7, executed for the last three governments in charge. The activity of the first two governments was mostly characterized by *covid-19*-related legislation, while the last government dedicated many efforts to work and economy-related topics (NRRP refers to the national economic recovery plan funded by the EU).

Table 7
Top 4 topics with the corresponding cardinality of the results of query Q7, executed for the last three governments in charge.

(a) Conte II Government		(b) Draghi I Government		(c) Meloni I Government	
Topic	**Count**	**Topic**	**Count**	**Topic**	**Count**
covid-19	783	covid-19	843	work	315
enterprises	510	work	609	safety	277
work	497	health	561	NRRP	221
health	440	enterprises	394	economy	183
…	…	…	…	…	…

Q5 *Ministries involvement in the legislative production*. By using the *domain* properties, we can derive statistics regarding the nature, timeliness, or even linguistic features of the laws produced by the same ministry under different governments. For instance, we can compute the frequency of signed laws by ministries for the last four Italian governments in charge. In Fig. 10, we propose a visual illustration of the query's output.

Q6 *EU Legislation Implemented in the Italian System*. An important component of annual reports is deriving statistics that characterize the source of each law in terms of being national legislation or deriving from the implementation of European Union legislation. Within our schema, we can directly query such results by leveraging the topics assigned to law nodes, searching for "eu regulation" or "eu directive", the two EU legislation types. For instance, the annual report of 2022-2023 (Osservatorio sulla legislazione della Camera dei Deputati, 2023) counted, for the first seven months of the XIX legislature, 13 laws that implemented EU legislation. We get 12 laws, missing only one entry, which we identified as a law containing modifications to previous EU implementation laws (i.e., law number 54/2023), thus not a direct implementation law.

Q7 *Topics of government intervention*, i.e., deriving the topics that characterize the government's approach in modifying or deleting previous legislation. We can track the topics of all the laws being amended/abrogated by a specific government and produce a straightforward visualization of the areas where the government has been most active (see Fig. 11 and Table 7).

6. Discussion on KG quality

In this section, we analyze the quality of the Knowledge Graph built by developing our pipeline. Given the domain of interest, we consider the following dimensions for evaluating the quality of our KG (in line with the surveys Wang et al., 2021; Xue & Zou, 2023):

1. **Accuracy**, measuring whether the KG correctly reflects the represented facts. Since there is no publicly available KG of the Italian legislation with the same granularity and timeliness, we can only measure accuracy by performing comparison experiments directly with the laws in their unstructured version. As a representative scenario to measure accuracy, we tested how effectively our KG handles the temporal dimension. To this aim, we focused on a specific timestamp (2023-12-31) and gathered all Italian laws in their updated version, i.e., with the text in force at that timestamp.[6] We programmatically analyzed the text of these laws and counted the *abrogated* ones, which are characterized by a textual abrogation formula within their text. Then, we compared such metrics with what can be inferred through our KG representation (see Q4), which only considers laws in their original version and allows us to infer temporal features, such as laws abrogations. This gave us

[6] As no API is available, the scraping of the dataset took around three days.

Table 8
Enrichments and status of KG nodes and their properties at the different steps of the ETL pipeline.

Node	AKN mapping	Data integration and error detection	LLM-enhancement
Law	74k nodes created no *domain* extracted no *topic* extracted	97% *domains* extracted via Parliamentary data	BERT *domain* extraction (3%) Mistral *topic* extraction (100%)
Article	310k nodes created 34% *titles* extracted no *topic* extracted	3k articles nodes with legal basis errors corrected	Mistral *title* extraction (66%) Mistral *topic* extraction (100%)
Attachment	216k nodes created no *titles* extracted no *topic* extracted		Mistral *title* extraction (100%) Mistral *topic* extraction (100%)
Government		68 nodes created	
Legislature		20 nodes created	

Table 9
Enrichments of KG edges at the different steps of the ETL pipeline.

Edge	AKN mapping	Data integration and error detection
Has Article	310k edges created	
Has Attachment	216k edges created	
Is Legal Basis of	100k edges created	6.3k cites corrections 145 inconsistency reported
Amends	82k edges created	3k incorrect source detected
Abrogates	65k edges created	1k incorrect source detected
Introduces	5k edges created	
Cites	235k edges created	
Under Government		74 K edges created
Under Legislature		74 K edges created
Succeeded By		67 edges created

the ratio of the stock of true abrogations that our query over the KG is capturing, equal to 0.98. Specifically, 109 laws out of 6283 "true abrogated laws" are not inferred through our KG. A manual inspection of such laws suggested that most of the edges in the *activeModification* tags were completely missing. In future work, we will explore how to handle such missing edges to achieve perfect accuracy. Nevertheless, the accuracy remains high, demonstrating how we can seamlessly capture the temporal dimension via graph queries.

2. **Completeness**, which refers to the degree to which all required information is present in the output of the ETL pipeline. Table 8 and Table 9 summarize how our pipeline enriches and improves the Knowledge Graph at each step. At the end of our pipeline, all nodes are enriched with all properties we defined in the schema, either through data integration (for domains) or via LLMs. Regarding edges, while we cannot measure the actual amount of missing ones – besides conducting experiments like the one deriving the accuracy of abrogation edges – we manage to correct and detect different types of errors, leading to a more complete graph.
3. **Consistency**, defined as the degree to which the knowledge of a KG does not contradict itself, i.e., defines no contradictions in the data concerning particular knowledge representation. In Section 4.3.2, we showed how we leverage graph queries that (i) detect errors in nodes and edges through heuristics, allowing us to adopt correction measures, and (ii) report material inconsistencies in the lawmaking activity, such as the case of articles that are cited but were repealed. In relative terms, the former represented only the 2% of all reference edges (`INTRODUCES`, `AMENDS`, `ABROGATES`, `CITES` edges), thus resulting in a KG with an overall high consistency, also considering our correction mechanisms.
4. **Timeliness**, i.e., the degree to which knowledge is up to date. The ETL pipeline can be run daily, constantly updating the KG with novel information. Therefore, we consider our pipeline to have high timeliness within the domain of legislative systems. The proposed PG schema allows us to focus only on novel laws and download only their original version, as temporal features are captured via graph queries, thus avoiding using a version (i.e., a new node) for each legislative "update". In Fig. 12, we tested the update time required for adding each law to the KG, and we compared it with the length of the law, measured in the number of words. We run our pipeline on our dedicated server machine with a 56-core Intel E5-2660 v4 CPU and 384 GB of RAM. Note that performances might vary according to multiple parameters, such as the number of citations, articles, and/or attachments, and the possibility of activating an LLM if required to complement a property; we chose the length of the law as an overall representative metric. In all cases, the execution time is very low; daily updates of the KG typically involve at most two or three laws. Nevertheless, we also computed the execution time required to re-create from scratch the entire graph of the Italian legislation since 1948, and, overall, it requires around 12 h, including the LLM calls.

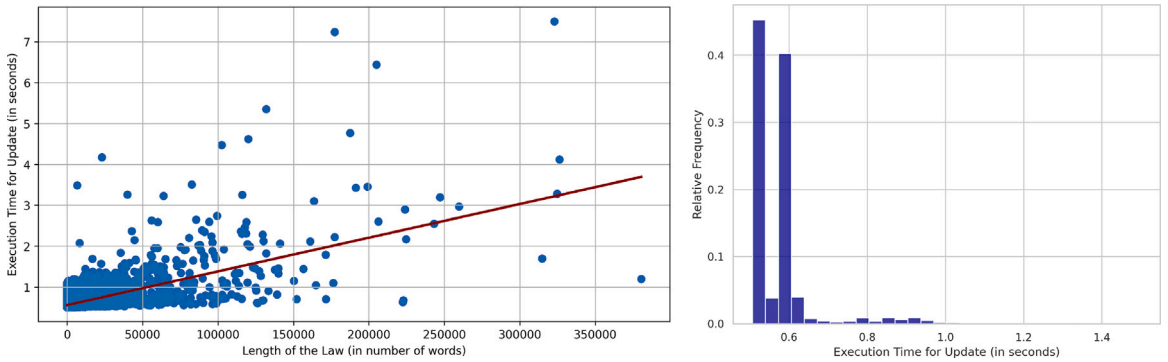


Fig. 12. On the left, a scatterplot representing the execution time, measured in seconds, required to process each law by running our ETL pipeline and compared to the textual length of the law. In red, we plot the line of a linear regression model, indicating the positive relationship between the length of the law and execution time. The maximum obtained value for the execution time of one law is 235 s, which is required to process the civil code, followed by the (very long) simplification decrees of 2008 and 2010 (all omitted from the plots for better visualization). On the right is the relative frequency distribution of execution times (truncated at 1.5 s for visualization purposes). The gap visible between the first and third bars is due to the LLM components, which are activated only in case of the integration of potential missing information.

5. **Trustworthiness**, i.e., the degree to which the information is accepted as correct and credible. In our KG construction, we use official data sources — such as the Official Gazette and the Italian Parliament. We have already discussed how our KG representation model can help improve the quality of the original data by reporting inconsistencies to the data source. In addition, our use of LLMs is task-specific, and by employing only open-source ones, their results and performance can be publicly scrutinized.
6. **Interoperability**, i.e., the degree to which the format and structure of the information conforms to data from other sources. The KG construction for the Italian legislation is based on the implementation of an internationally adopted standard (AKN) within the national system, meaning that, for countries that adopt the same standard, the ETL pipeline is fully replicable, with national-specific adaptations for the different national data sources.

7. Conclusion

In this paper, we presented an end-to-end pipeline starting from a recently adopted international standard, Akoma Ntoso, and via a controlled use of LLMs, constructs a high-quality knowledge graph of the Italian legislation. We proposed a graph schema based on the property graph paradigm and the recently standardized Graph Query Language, which is designed to efficiently represent legislative systems and to capture complex law-related aspects, such as the temporal dimension. To the best of our knowledge, the ETL pipeline is the first to combine the recently adopted XML machine-readable standard for creating the graph, Akoma Ntoso, and a property graph, GQL-compliant approach. Given the international adoption of this standard, we believe that the same pipeline can be readily adapted for use in other systems with minimal adjustments for country-specific attributes. As a result, information obtained from various legislative graphs could potentially be comparable, facilitating the extraction of further insights. We expanded the graph completeness by leveraging LLMs to derive or complement the nodes' properties. To this aim, we also focused on fine-tuning sufficiently light models that allow us to reduce the computational requirements and to achieve the information extraction tasks comparably well to state-of-the-art language models. We also explored how this model and its enhancement allowed us to derive insights into the legislative system, allowing the automation of manually computed statistics and expanding it to novel, valuable metrics. Finally, we discussed the overall high quality of the KG across multiple dimensions. In particular, we demonstrated its accuracy and efficiency in capturing the temporal dimension and showed its overall high consistency and completeness, also thanks to the integration of LLM components. Our pipeline resolves one sufficiently complex example (the Italian case) in the legislation field and shows successful results; with this, we aim to pave the way for easy-to-drive knowledge management of legislative systems, possibly also allowing inter-system comparisons.

CRediT authorship contribution statement

Andrea Colombo: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Data curation, Conceptualization. **Anna Bernasconi:** Writing – review & editing, Supervision, Conceptualization. **Stefano Ceri:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the PNRR-PE-AI FAIR project, funded by the NextGenerationEU program. Andrea Colombo kindly acknowledges INPS for funding his Ph.D. program.

Appendix

A.1. Formal property graph schema

The formal definition of the Property Graph Schema (Angles et al., 2023) proposed for the KG is:

```
CREATE GRAPH TYPE lawsGraphType STRICT{
 (lawType: Law {id STRING, title STRING, typeLaw STRING, publicationDate DATE, inForceDate
  DATE, numArt INT,numAttach INT, domain LIST, topic LIST}),
 (articleType: Article {id STRING, title STRING, number INT, text STRING, topic LIST}),
 (attachmentType: Attachment {id STRING, title STRING, type STRING, text STRING, topic LIST}),
 (legislatureType: Legislature {name STRING, startDate DATE, endDate DATE}),
 (governmentType: Government {name STRING, startDate DATE, endDate DATE}),
 (:lawType)-[hasArticleType: has_article]->(:articleType),
 (:lawType)-[hasAttachmentType: has_attachment]->(:attachmentType),
 (:lawType)-[underGovernmentType: under_government]->(:governmentType),
 (:lawType)-[underLegislatureType: under_legislature]->(:legislatureType),
 (:governmentType)-[succededByType: succeded_by]->(:governmentType),
 (:lawType)-[referenceType: is_legal_basis_of {paragraph LIST, weight INT}]->(:lawType),
 (:articleType)-[referenceType: is_legal_basis_of|cites {paragraph LIST,
  weight INT}]->(:lawType),
 (:articleType)-[referenceType: amends|introduces|abrogates {paragraph LIST, newText STRING}]
  ->(:lawType),
 (:articleType)-[referenceType: amends|introduces|abrogates {paragraph LIST, newText STRING}}]
  ->(:articleType),
 (:articleType)-[referenceType: amends|introduces|abrogates {paragraph LIST, newText STRING}}]
  ->(:attachmentType),
 (:articleType)-[referenceType:  cites {paragraph LIST, weight INT}]->(:articleType),
 (:articleType)-[referenceType:  cites {paragraph LIST, weight INT}]->(:attachmentType),
 (:attachmentType)-[referenceType: is_legal_basis_of|cites {paragraph LIST, weight INT}]->(:lawType),
 (:attachmentType)-[referenceType: cites {paragraph LIST, weight INT}]->(:articleType),
 (:attachmentType)-[referenceType: cites {paragraph LIST, weight INT}]->(:attachmentType)}
```

A.2. Queries capturing the temporal dimension

In this section, we report two relevant temporal-dependent Cypher queries that illustrate the capability of our schema to capture the evolution of the legislative corpus. The first query derives the text of a law in force in a certain point in time, as modified by the legislation. For instance, to derive law 14/2010 as it was in force at timestamp 2023-02-01:

```
CALL{
  MATCH (l:Law)-[:HAS_ARTICLE]->(a:Article)
  OPTIONAL MATCH (a)<-[r:ABROGATES|AMENDS|INTRODUCES]-(a2)<-[:HAS_ARTICLE]-(l2:Law)
  WHERE l2.publicationDate < datetime("2010|14")
  WITH l.id AS IDLAW, a.id AS IDART, a.number AS NUMART, MAX(l2.publicationDate)
    AS LASTMOD
  WHERE IDLAW = "2010|14"
  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

  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 = "2010|14"
   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("2010|14")
   WITH IDLAW, IDART, a.number AS NUMART, a.text AS TEXT, NCHANGES,
      COUNT(*) AS FUTURECHANGES
   WHERE NCHANGES = FUTURECHANGES
   RETURN IDLAW, IDART, TEXT, NUMART

   UNION

   MATCH (l:Law)-[:HAS_ARTICLE]->(a:Article)
   WHERE NOT (a)<-[:ABROGATES|AMENDS|INTRODUCES]-() AND l.id = "2010|14"
   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)
```

The second Cypher query infers the updated list of laws that are abrogated:

```
MATCH p=(l:Law)-[:HAS_ARTICLE]->(a:Article)<-[r:ABROGATES]-(a2:Article)<-
 [:HAS_ARTICLE]-(l2:Law) WHERE r.paragraph IS NULL
WITH l.id AS abrogatedLaw, l.numArt AS N_Arts, COUNT(DISTINCT a)
   AS N_Repeals WHERE N_Repeals >= N_Arts
WITH COLLECT(abrogatedLaw) AS list_abrogations
MATCH (l:Law) WHERE l.id IN list_abrogations RETURN l.id
UNION
MATCH p=(l:Law)<-[r:ABROGATES]-() WHERE r.paragraph  IS NULL RETURN l.id
```

A.3. Parliamentary data integration

From the Camera dei Deputati endpoint (Camera dei Deputati, 2024), we can derive the historical names of all departments throughout the Italian Republic with the following SPARQL query:

```
SELECT DISTINCT ?titolo
WHERE {
 ?governo rdf:type ocd:governo .
 ?governo dc:title ?Name.
 ?governo ocd:startDate ?Start .
 OPTIONAL {?governo ocd:endDate ?End.}
 ?governo ocd:rif_membroGoverno ?membro .
 ?membro foaf:surname ?cognome; dc:title ?titolo .
}
```

A.4. Italian domain list and dictionary

We report the list of 16 domain in Italian that we consider throughout the paper: interno, istituzioni, agricultura, istruzione, economia, comunicazioni, presidenza, trasporti, sanità, esteri, giustizia, lavoro, difesa, pubblica amministrazione, cultura e ambiente, sport e turismo.

A sample of the domain dictionary mapping is available in Table A.1.

A.5. Details about LLMs fine-tuning

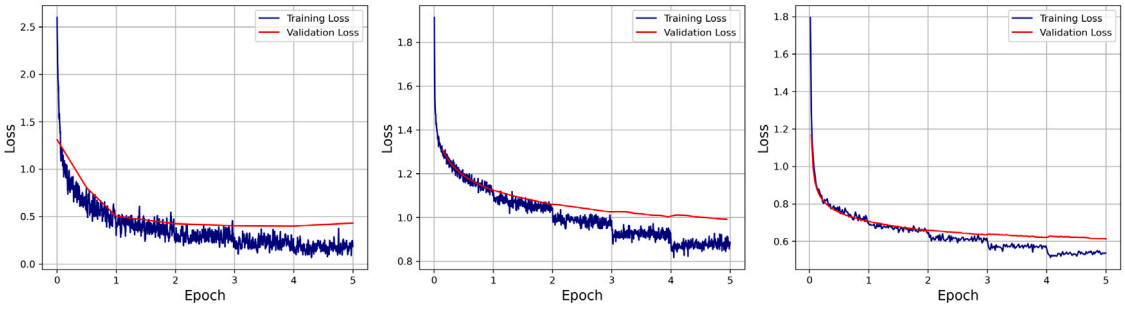
Training losses and parameters used for fine-tuning the LLMs are available in Fig. A.1 and Table A.2.

A.6. Title and topic extraction via LLM - Italian versions

Since the LLMs are used in Italian, we depict the original versions for Figs. 7, 8 and Tables 3, 4 (see Figs. A.2 and A.3 and Tables A.3 and A.4).

Table A.1
Italian domain dictionary mapping keywords to a domain.

Keyword in ministry name	Domain
aviazione	trasporti
parlamento	istituzioni
affari esteri	esteri
previdenza sociale	lavoro
foreste	agricoltura
...	...



(a) BERT Domain Classifier (b) Mistral-7B Title Extractor (c) Mistral-7B Topic Extractor

Fig. A.1. Train and validation loss of the fine-tuning steps for the BERT domain classifier model, the Mistral-7B title extractor and the Mistral-7B topic extractor. For all of these models, the validation loss starts to converge around the 4th epoch.

Table A.2
Parameters used for fine-tuning the BERT domain classifier model, the Mistral-7B title extractor and the Mistral-7B topic extractor.

(a) BERT domain classifier		(b) Mistra-7B title extractor		(c) Mistral-7B topic extractor	
Parameter	Value	Parameter	Value	Parameter	Value
Batch size	32	Batch size	4	Batch size	4
Learning rate	2e−5	Learning rate	1e−4	Learning rate	1e−4
Training time	40 min.	Training time	9 h	Training time	9 h
Best validation loss	0.45	Best validation loss	1.003	Best validation loss	0.61
Accuracy	0.90	Optimization	Adam	Optimization	Adam
Optimization	AdamW	Training set size	108k	Training set size	74k
Training set size	45k	Warm-up fraction	0.03	Warm-up fraction	0.03
		Fine-tuning technique	LoRa	Fine-tuning technique	LoRa

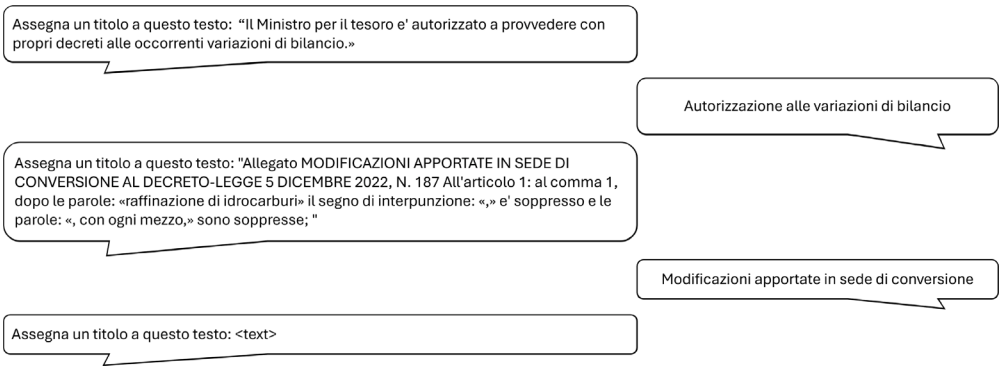


Fig. A.2. Few-shot learning for Title Extraction Task (Italian version).

A.7. Knowledge graph exploration - Cypher queries

In this section, we report the Cypher queries used to derive the insights of Section 5:

1. Laws enacted per year (Q1):

Table A.3
Examples of title extraction pre and post-fine tuning (Italian versions).

Text	Mistral-7B Pre-Trained	Mistral-7B Fine-tuned
Articolo 13. 1. Il presente Protocollo entra in vigore alla data concordata tra le Parti con successivo scambio di note. 2. Il Protocollo resta in vigore per 5 anni. Salvo che una delle due Parti comunichi, con preavviso di almeno sei mesi rispetto alla scadenza, la propria intenzione di non rinnovare...	Articolo 13	Disposizioni finali e termini di vigore del Protocollo
Art. 7. 1. Il regime previsto dalla legge 1 dicembre 1948, n. 1438 , e successive modificazioni, limitatamente al prodotto contraddistinto con il n. 13 della tabella A, allegata alla legge 27 dicembre 1975, n. 700, è rideterminato secondo quanto previsto nel presente articolo, in conformità agli articoli 30 e 32 dell'accordo di cooperazione tra la Comunità economica europea e la Repubblica di Slovenia...	Modifiche al regime di agevolazioni per il prodotto contraddistinto con il n. 13 della tabella A	Modifiche al regime di agevolazioni per il carburante in Slovenia
Art. 33. 1.Le organizzazioni di volontariato possono assumere lavoratori dipendenti o avvalersi di prestazioni di lavoro auto- nomo o di altra natura esclusivamente nei limiti necessari al loro regolare funzionamento oppure nei limiti occorrenti a qualificare...	Articolo 33. Organizzazioni di volontariato	Finanziamento delle organizzazioni di volontariato

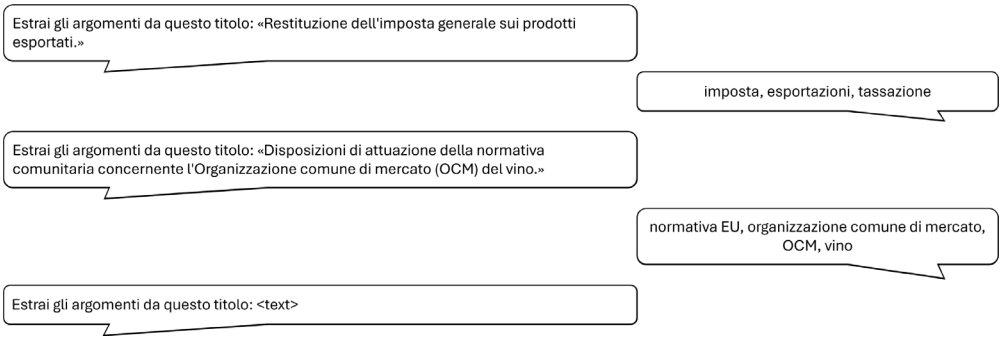


Fig. A.3. Few-shot learning for Topic Extraction Task (Italian version).

Table A.4
Examples of topic extraction pre and post-fine tuning (Italian version).

Title	Mixtral-8 × 22B Pre-Trained	Mistral-7B Fine-tuned
Caratteristiche dell'aumento di capitale	capitale, aumento, caratteristiche	aumento di capitale, finanza aziendale
Entrata in vigore	entrata in vigore, attivazione, attivazione normativa, attivazione regolare, attivazione	entrata in vigore
Aumento del contributo personale annuo	contributo personale, anno, aumento	contributo personale, pensioni
Poteri di controllo e perquisizione delle forze di polizia	polizia, controllo, perquisizione, poteri	polizia, controllo, perquisizione

```
MATCH (l:Law)
RETURN l.publicationDate.year AS Date, count(l) AS Num
```

2. Laws never cited after publication (Q2):

```
MATCH (l:Law)-[:HAS_ART|HAS_ATTACHMENT]->(a)
WHERE NOT (l)-[:IS_LEGAL_BASIS_OF]->(:Law)
AND NOT (a)-[:IS_LEGAL_BASIS_OF]->(:Law)
AND NOT (a)<-[:AMENDS]-() AND NOT (a)<-[:ABROGATES]-()
AND NOT (a)<-[:CITES]-() AND NOT (a)<-[:INTRODUCES]-()
AND NOT (l)<-[:AMENDS]-() AND NOT (l)<-[:ABROGATES]-() AND NOT (l)<-[:CITES]-()
RETURN l.publicationDate.year as Date, COUNT(DISTINCT l)
```

3. Outdated laws (Q3):

```
MATCH (l:Law)-[:IS_LEGAL_BASIS_OF]->(l2:Law)
WHERE l2.publicationDate < datetime("1960")
WITH COLLECT(l.id) AS CitedBefore60s
MATCH (l:Law)-[:IS_LEGAL_BASIS_OF]->(l2:Law)
```

```
WHERE l2.publicationDate > datetime("1990")
AND l.id IN CitedBefore60s
WITH COLLECT(l.id) AS StillCited, CitedBefore60s
UNWIND[x IN CitedBefore60s WHERE NOT
  ANY(z IN StillCited WHERE z CONTAINS x)] AS OutdatedLaws
RETURN OutdatedLaws
```

4. Stock of in-force laws (Q4):

```
MATCH p=(l:Law)-[:HAS_ARTICLE]->(a:Article)<-[r:ABROGATES]-(a2:Article)<-
    [:HAS_ARTICLE]-(l2:Law)
WHERE r.paragrah IS NULL AND l2.publicationDate <= datetime('2020')
WITH l.id AS abrogatedLaw, l.numArt AS N_Arts, COUNT(DISTINCT a)
   AS N_Repeals
WHERE N_Repeals >= N_Arts
WITH COLLECT(abrogatedLaw) AS list_abrogations
MATCH (l:Law)
WHERE l.publicationDate <= datetime('2020') AND NOT l.id IN
  list_abrogations AND NOT ()-[:ABROGATES]->(l:Law)
RETURN COUNT(l.id) AS CountInForceLaws
```

5. Ministries involvement in the legislative production (Q5):

```
MATCH (g:Government)-[:SUCCEDED_BY*4]->(g2:Government)
WHERE NOT EXISTS ((g2)-[:SUCCEDED_BY]->(:Government))
WITH g.name AS FIRSTGOV
MATCH (g:Government)-[:SUCCEDED_BY*1..4]->(g2:Government)
MATCH (l:Law)-[:UNDER_GOVERNMENT]-(g2:Government)
WHERE g.name = FIRSTGOV
WITH g2.name AS GOVNAME, COUNT(l) AS NLAWS
MATCH (l:Law)-[:UNDER_GOVERNMENT]-(g2:Government)
WHERE g2.name = GOVNAME
WITH g2.name AS GOVNAME, NLAWS, l.domain as allDomains
UNWIND allDomains as DOMAIN
RETURN GOVNAME, NLAWS, DOMAIN, COUNT(DOMAIN) AS N
```

6. EU Legislation Implemented in the Italian System (Q6):

```
MATCH (l:Law)-[:UNDER_LEGISLATURE]->(e:Legislature)
WHERE ANY(x IN l.topic WHERE x IN ["direttiva ue","regolamento ue"]) AND e.name =
"Legislatura XIX" AND l.publicationDate <= e.startDate + Duration({months: 7})
RETURN COUNT(l) as EUConversions
```

7. Topics of government intervention (Q7)

```
MATCH (l1:Law)-[:HAS_ARTICLE]->(a1)<-[:ABROGATES|AMENDS|INTRODUCES]-(a2)
    <-[:HAS_ARTICLE]-(l2:Law)-[:UNDER_GOVERNMENT]->(g:Government)
WHERE  g.name = "I Governo Meloni"
UNWIND l1.topic as topics
RETURN g.name AS GOVNAME, topics AS TOPIC, COUNT(topics) as N
```

Data availability

Data is shared in a public repository. AI models used are also shared on HuggingFace.

References

Anelli, V. W., Brienza, E., Recupero, M., Greco, F., Maria, A. D., Di Noia, T., et al. (2023). Navigating the legal landscape: Developing Italy's official legal knowledge graph for enhanced legislative and public services. In F. Falchi, F. Giannotti, A. Monreale, C. Boldrini, S. Rinzivillo, S. Colantonio (Eds.), *CEUR workshop proceedings: vol. 3486, Proceedings of the Italia intelligenza artificiale - thematic workshops co-located with the 3rd CINI national lab AIIS conference on artificial intelligence (ital IA 2023), pisa, Italy, May 29-30, 2023* (pp. 223–228). CEUR-WS.org, URL: https://ceur-ws.org/Vol-3486/87.pdf.

Angelidis, I., Chalkidis, I., Nikolaou, C., Soursos, P., & Koubarakis, M. (2018). Nomothesia: A linked data platform for greek legislation. In *Proceedings of MIREL 2018 workshop on mIning and rEasoning with legal texts, Luxembourg, 30-08-2018*. URL: https://cgi.di.uoa.gr/~koubarak/publications/2018/nomothesia-linked-data.pdf.

Angles, R. (2018). The property graph database model. In D. Olteanu, & B. Poblete (Eds.), *CEUR workshop proceedings: vol. 2100, Proceedings of the 12th alberto mendelzon international workshop on foundations of data management, cali, Colombia, May 21-25, 2018*. CEUR-WS.org, URL: https://ceur-ws.org/Vol-2100/paper26.pdf.

Angles, R., Arenas, M., Barceló, P., Hogan, A., Reutter, J., & Vrgoč, D. (2017). Foundations of modern query languages for graph databases. *ACM Computing Surveys*, *50*(5), http://dx.doi.org/10.1145/3104031.

Angles, R., Bonifati, A., Dumbrava, S., Fletcher, G., Green, A., Hidders, J., et al. (2023). PG-schema: Schemas for property graphs. *Proceedings ACM Management Data*, *1*(2), http://dx.doi.org/10.1145/3589778.

Athan, T., Boley, H., Governatori, G., Palmirani, M., Paschke, A., & Wyner, A. (2013). OASIS LegalRuleML. In *Proceedings of the fourteenth international conference on artificial intelligence and law* (pp. 3–12). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/2514601.2514603.

Barabucci, G., Cervone, L., Palmirani, M., Peroni, S., & Vitali, F. (2009). Multi-layer markup and ontological structures in akoma ntoso. In *International workshop on AI approaches to the complexity of legal systems* (pp. 133–149). Beijing, China: Springer, Springer, http://dx.doi.org/10.1007/978-3-642-16524-5_9.

Camera dei Deputati (2024). SPARQL endpoint. URL: https://dati.camera.it/sparql.

Chen, Q., Du, J., Allot, A., & Lu, Z. (2022). LitMC-BERT: transformer-based multi-label classification of biomedical literature with an application on COVID-19 literature curation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *19*(5), 2584–2595. http://dx.doi.org/10.1109/TCBB.2022.3173562.

Ciglan, M., Averbuch, A., & Hluchy, L. (2012). Benchmarking traversal operations over graph databases. In *2012 IEEE 28th international conference on data engineering workshops* (pp. 186–189). http://dx.doi.org/10.1109/ICDEW.2012.47.

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

Colombo, A. (2024a). BERT-DomainItalianLaws. http://dx.doi.org/10.57967/hf/2639, URL: https://huggingface.co/andre156/BERT-DomainLaws.

Colombo, A. (2024b). [Dataset] knowledge graph of the Italian legislation. http://dx.doi.org/10.5281/zenodo.13375511.

Colombo, A. (2024c). Italian-laws-title-extraction Mistral7B fine tuned model. http://dx.doi.org/10.57967/hf/2928, URL: https://huggingface.co/andre156/italian-laws-title-extraction.

Colombo, A. (2024d). Italian-laws-topic-extraction Mistral7B fine tuned model. http://dx.doi.org/10.57967/hf/2929, URL: https://huggingface.co/andre156/italian-laws-topic-extraction.

Crotti Junior, A., Orlandi, F., Graux, D., Hossari, M., O'Sullivan, D., Hartz, C., et al. (2020). Knowledge graph-based legal search over german court cases. In *The semantic web: ESWC 2020 satellite events: ESWC 2020 satellite events, heraklion, crete, Greece, May 31 – June 4, 2020, revised selected papers* (pp. 293–297). Berlin, Heidelberg: Springer-Verlag, http://dx.doi.org/10.1007/978-3-030-62327-2_44.

Crotti Junior, A., Orlandi, F., O'Sullivan, D., Dirschl, C., & Reul, Q. (2019). Using mapping languages for building legal knowledge graphs from XML files. In R. Samavi, M. P. Consens, S. Khatchadourian, V. Nguyen, A. P. Sheth, J. M. Giménez-García, & H. Thakkar (Eds.), *CEUR workshop proceedings: vol. 2599, Proceedings of the blockchain enabled semantic web workshop (blockSW) and contextualized knowledge graphs (CKG) workshop co-located with the 18th international semantic web conference, blockSW/cKG@iSWC 2019, auckland, New zealand, October 27, 2019*. CEUR-WS.org, URL: https://ceur-ws.org/Vol-2599/CKG2019_paper_6.pdf.

Curtotti, M., & McCreath, E. (2012). Enhancing the visualization of law. *vol. 9*, In *Law via the internet twentieth anniversary conference, Cornell university, October*. http://dx.doi.org/10.2139/ssrn.2160614.

Curtotti, M., McCreath, E., & Sridharan, S. (2013). Software tools for the visualization of definition networks in legal contracts. In *Proceedings of the fourteenth international conference on artificial intelligence and law* (pp. 192–196). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/2514601.2514625.

Das, S., Srinivasan, J., Perry, M., Chong, E. I., & Banerjee, J. (2014). A tale of two graphs: Property graphs as RDF in oracle. In *Proceedings of the 17th international conference on extending database technology, athens, Greece, March 24–28*. http://dx.doi.org/10.5441/002/edbt.2014.82.

de Maat, E., Winkels, R., & van Engers, T. (2006). Automated detection of reference structures in law. In *Proceedings of the 2006 conference on legal knowledge and information systems: JURIX 2006: the nineteenth annual conference* (pp. 41–50). NLD: IOS Press, http://dx.doi.org/10.5555/1563577.1563583.

Deutsch, A., Francis, N., Green, A., Hare, K., Li, B., Libkin, L., et al. (2022). Graph pattern matching in GQL and SQL/pgq. In *Proceedings of the 2022 international conference on management of data* (pp. 2246–2258). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3514221.3526057.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805. arXiv:1810.04805. URL: http://arxiv.org/abs/1810.04805.

Dong, X. L. (2023). Generations of knowledge graphs: The crazy ideas and the business impact. *Proceedings of VLDB Endowment*, *16*(12), 4130–4137. http://dx.doi.org/10.14778/3611540.3611636.

EIEF (2024). Website of einaudi institute for economics and finance. Available online at: https://www.eief.it/eief//. (last Accessed 01 May 2024).

European Union Publications Office (2023). A common structured format for EU legislative documents. Available online at: https://op.europa.eu/it/web/eu-vocabularies/akn4eu. (last Accessed 01 May 2024).

European Union Publications Office (2024). About ELI - EUR-lex. https://eur-lex.europa.eu/eli-register/about.html. (Accessed 15 May 2024).

Flatt, A., Langner, A., & Leps, O. (2022). *Model-driven development of akoma ntoso application profiles: a conceptual framework for model-based generation of xml subschemas*. Springer International Publishing, http://dx.doi.org/10.1007/978-3-031-14132-4.

Francis, N., Gheerbrant, A., Guagliardo, P., Libkin, L., Marsault, V., Martens, W., et al. (2023). A researcher's digest of GQL. In *The 26th international conference on database theory, 2023*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, http://dx.doi.org/10.4230/LIPIcs.ICDT.2023.1, 1–1.

Francis, N., Green, A., Guagliardo, P., Libkin, L., Lindaaker, T., Marsault, V., et al. (2018). Cypher: An evolving query language for property graphs. In *Proceedings of the 2018 international conference on management of data* (pp. 1433–1445). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3183713.3190657.

Giommoni, T., Morelli, M., & Paserman, M. D. (2022). Signalling incentives and the quality of legislation: A text and network analysis of the US congress. In *The wallis institute annual conference*. URL: https://www.wallis.rochester.edu/assets/pdf/conference29/signaling-incentives-and-quality-of-legislation.pdf.

Guia, J., Soares, V. G., & Bernardino, J. (2017). Graph databases: Neo4j analysis. In *Proceedings of the 19th international conference on enterprise information systems - volume 1: ICEIS* (pp. 351–356). INSTICC, SciTePress, http://dx.doi.org/10.5220/0006356003510356.

Gunasekar, S., Zhang, Y., Aneja, J., Mendes, C. C. T., Del Giorno, A., Gopi, S., et al. (2023). Textbooks are all you need. arXiv preprint arXiv:2306.11644.

He, P., Huang, J., & Li, M. (2024). Text keyword extraction based on GPT. In *2024 27th international conference on computer supported cooperative work in design* (pp. 1394–1398). IEEE, http://dx.doi.org/10.1109/CSCWD61410.2024.10580849.

Hoekstra, R., Breuker, J., Marcello, D. B., & Boer, A. (2007). The LKIF core ontology of basic legal concepts. *CEUR Workshop Proceedings*, *321*, 43–63, URL: https://ceur-ws.org/Vol-321/paper3.pdf, 2nd Workshop on Legal Ontologies and Artificial Intelligence Techniques, LOAIT 2007 ; Conference date: 04-06-2007 Through 04-06-2007.

Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G. D., Gutierrez, C., et al. (2021). Knowledge graphs. *ACM Computing Surveys*, *54*(4), http://dx.doi.org/10.1145/3447772.

Honnibal, M., Montani, I., Landeghem, S. V., & Boyd, A. (2020). Spacy: Industrial-strength natural language processing in python. URL: https://spacy.io/, Version 2.3.2.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., et al. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

Invernici, F., Bernasconi, A., & Ceri, S. (2024). Exploring the evolution of research topics during the COVID-19 pandemic. *Expert Systems with Applications*, *252*, 124028. http://dx.doi.org/10.1016/j.eswa.2024.124028.

ISO (2024). ISO/IEC 39075:2024 - information technology — Database languages — GQL. URL: https://www.iso.org/standard/76120.html.

Istituto Poligrafico e Zecca dello Stato (2024). Normattiva website. https://www.normattiva.it/ricerca/avanzata. (Accessed 15 May 2024).

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., et al. (2023). Mistral 7B. arXiv preprint arXiv:2310.06825.

Karpen, U. (2008). Instructions for law drafting. *European Journal of Law Reform*, *10*, 163–181, URL: https://www.elevenjournals.com/tijdschrift/ejlr/2008/2/EJLR_1387-2370_2008_010_002_004.pdf.

Kukreja, S., Kumar, T., Purohit, A., Dasgupta, A., & Guha, D. (2024). A literature survey on open source large language models. In *Proceedings of the 2024 7th international conference on computers in management and business* (pp. 133–143). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3647782.3647803.

Legix. Info (2012). Applying akoma ntoso to the United States code. Available online at: https://xcential.com/blog/applying-akoma-ntoso-to-the-united-states-code/. (last Accessed 01 May 2024).

Libkin, L., Martens, W., & Vrgoč, D. (2016). Querying graphs with data. *Journal of the ACM*, *63*(2), http://dx.doi.org/10.1145/2850413.

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

Lupo, C., Vitali, F., Francesconi, E., Palmirani, M., Winkels, R., de Maat, E., et al. (2007). *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.

Mihindukulasooriya, N., Tiwari, S., Enguix, C. F., & Lata, K. (2023). Text2KGBench: A benchmark for ontology-driven knowledge graph generation from text. In T. R. Payne, V. Presutti, G. Qi, M. Poveda-Villalón, G. Stoilos, L. Hollink, Z. Kaoudi, G. Cheng, & J. Li (Eds.), *The semantic web – ISWC 2023* (pp. 247–265). Cham: Springer Nature Switzerland, URL: https://api.semanticscholar.org/CorpusID:260611736.

Moreno-Schneider, J., Rehm, G., Montiel-Ponsoda, E., Rodriguez-Doncel, V., Revenko, A., Karampatakis, S., et al. (2020). Orchestrating NLP services for the legal domain. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (Eds.), *Proceedings of the twelfth language resources and evaluation conference* (pp. 2332–2340). Marseille, France: European Language Resources Association, URL: https://aclanthology.org/2020.lrec-1.284.

Mu, Y., Dong, C., Bontcheva, K., & Song, X. (2024). Large language models offer an alternative to the traditional approach of topic modelling. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-cOLING 2024)* (pp. 10160–10171). http://dx.doi.org/10.48550/arXiv.2403.16248.

Neo4J (2024). Cypher query language documentation. Available online at: https://neo4j.com/docs/cypher-manual/current/introduction/. (last Accessed 01 May 2024).

OASIS (2018). Akoma ntoso version 1.0 becomes an OASIS standard. Available online at: https://www.oasis-open.org/news/announcements/akoma-ntoso-version-1-0-becomes-an-oasis-standard/. (last Accessed 01 May 2024).

Oliveira, F. d., & Oliveira, J. M. P. d. (2023). A RDF-based graph to representing and searching parts of legal documents. *Artificial Intelligence and Law*, http://dx.doi.org/10.1007/s10506-023-09364-9.

OpenAI (2024). GPT-4 technical report. arXiv:2303.08774, URL: https://arxiv.org/abs/2303.08774.

Orlandi, F., Graux, D., & O'Sullivan, D. (2021). Benchmarking RDF metadata representations: Reification, singleton property and RDF. In *2021 IEEE 15th international conference on semantic computing* (pp. 233–240). http://dx.doi.org/10.1109/ICSC50631.2021.00049.

Osservatorio sulla legislazione della Camera dei Deputati (2023). La legislazione tra stato, regioni e unione Europea rapporto 2022–2023. URL: https://www.camera.it/temiap/2023/11/20/OCD177-6772.pdf.

Palmirani, M. (2018). Akoma ntoso for making FAO resolutions accessible. In G. Peruginelli, & S. Faro (Eds.), *Frontiers in artificial intelligence and applications*: *vol. 317*, *Knowledge of the law in the big data age, conference 'law via the internet 2018', florence, Italy, 11-12 October 2018* (pp. 159–169). IOS Press, http://dx.doi.org/10.3233/FAIA190018.

Palmirani, M. (2021). Lexdatafication: Italian legal knowledge modelling in akoma ntoso. In V. Rodríguez-Doncel, M. Palmirani, M. Araszkiewicz, P. Casanovas, U. Pagallo, & G. Sartor (Eds.), *AI approaches to the complexity of legal systems XI-XII* (pp. 31–47). Cham: Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-89811-3_3.

Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., & Wu, X. (2024). Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, *36*(7), 3580–3599. http://dx.doi.org/10.1109/TKDE.2024.3352100.

Parnami, A., & Lee, M. (2022). Learning from few examples: A summary of approaches to few-shot learning. arXiv preprint arXiv:2203.04291.

Pérez, J., Arenas, M., & Gutierrez, C. (2009). Semantics and complexity of SPARQL. *ACM Transactions on Database Systems*, *34*(3), 1–45. http://dx.doi.org/10.1145/1567274.1567278.

Purpura, S., & Hillard, D. (2006). Automated classification of congressional legislation. In *Dg.o '06*, *Proceedings of the 2006 international conference on digital government research* (pp. 219–225). Digital Government Society of North America, http://dx.doi.org/10.1145/1146598.1146660.

Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M., Sakib, S., Mim, M. M. J., et al. (2024). A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*, *12*, 26839–26874. http://dx.doi.org/10.1109/ACCESS.2024.3365742.

Ray, P. P. (2023). ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, *3*, 121–154. http://dx.doi.org/10.1016/j.iotcps.2023.04.003.

Robinson, I., Webber, J., & Eifrem, E. (2015). *Graph databases:new opportunities for connected data* (2nd ed.). O'Reilly Media, Inc., http://dx.doi.org/10.5555/2846367.

Rodríguez-Doncel, V., Navas-Loro, M., Montiel-Ponsoda, E., & Casanovas, P. (2018). Spanish legislation as linked data.. In *CEUR workshop proceedings*, *Proceedings of the 2nd workshop on technologies for regulatory compliance co-located with the 31st international conference on legal knowledge and information systems (JURIX 2018), groningen, the netherlands, December 12, 2018* (pp. 135–141). Groningen, The Netherlands: CEUR-WS.org, URL: https://ceur-ws.org/Vol-2309/12.pdf.

Sadeghian, A., Sundaram, L., Wang, D. Z., Hamilton, W. F., Branting, K., & Pfeifer, C. (2018). Automatic semantic edge labeling over legal citation graphs. *Artificial Intelligence and Law*, *26*(2), 127–144. http://dx.doi.org/10.1007/s10506-018-9217-1.

Sana, A., & Suganthi, G. (2017). Modeling and storage of XML data as a graph and processing with graph processor. *World Congress on Computing and Communication Technologies (WCCCT)*, 16–19. http://dx.doi.org/10.1109/WCCCT.2016.14.

Sansone, C., & Sperlí, G. (2022). Legal information retrieval systems: State-of-the-art and open issues. *Information Systems*, *106*, Article 101967. http://dx.doi.org/10.1016/j.is.2021.101967, URL: https://www.sciencedirect.com/science/article/pii/S0306437921001551.

Seaborne, A. (2013). SPARQL 1.1 Query Results CSV and TSV Formats. Available online at: https://www.w3.org/TR/sparql11-results-csv-tsv/. (last Accessed 01 May 2024).

Senato della Repubblica (2001). Regole e raccomandazioni per la formulazione tecnica dei testi legislativi. Available online at: https://www.senato.it/istituzione/circolari-del-presidente/regole-raccomandazioni-formulazione-tecnica-testi-legislativi.

Solid IT consulting (2024). DB-Engines Ranking of Graph DBMS. Available online at: https://db-engines.com/en/ranking/graph+dbms. (last Accessed 01 May 2024).

UN System Chief Executives Board for Coordination (2017). Akoma Ntoso for the United Nations. Available online at: https://unsceb.org/unsif-akn4un. (last Accessed 01 May 2024).

US Library of Congress (2024). Congress.gov application programming interface (API). Available online at: https://github.com/LibraryOfCongress/api.congress.gov/.

Vitali, F., Palmirani, M., et al. (2019). Akoma ntoso: flexibility and customization to meet different legal traditions. *Balisage Series on Markup Technologies*, *24*, 1–9. http://dx.doi.org/10.4242/BalisageVol24.Palmirani01.

Wadhwa, S., Amir, S., & Wallace, B. C. (2023). Revisiting relation extraction in the era of large language models. *vol. 2023*, In *Proceedings of the conference. association for computational linguistics. meeting* (p. 15566). NIH Public Access, http://dx.doi.org/10.18653/v1/2023.acl-long.868.

Wang, X., Chen, L., Ban, T., Usman, M., Guan, Y., Liu, S., et al. (2021). Knowledge graph quality control: A survey. *Fundamental Research*, *1*(5), 607–626. http://dx.doi.org/10.1016/j.fmre.2021.09.003.

Wang, J., Huang, J. X., & Sheng, J. (2024). An efficient long-text semantic retrieval approach via utilizing presentation learning on short-text. *Complex & Intelligent Systems*, *10*(1), 963–979. http://dx.doi.org/10.1007/s40747-023-01192-3.

Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, *53*(3), http://dx.doi.org/10.1145/3386252.

Wu, N., Gong, M., Shou, L., Liang, S., & Jiang, D. (2023). Large language models are diverse role-players for summarization evaluation. In *CCF international conference on natural language processing and Chinese computing* (pp. 695–707). Springer, http://dx.doi.org/10.1007/978-3-031-44693-1_54.

Wu, C.-J., Raghavendra, R., Gupta, U., Acun, B., Ardalani, N., Maeng, K., et al. (2022). Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*, *4*, 795–813. http://dx.doi.org/10.48550/arXiv.2111.00364.

Wulczyn, E., Khabsa, M., Vora, V., Heston, M., Walsh, J., Berry, C., et al. (2016). Identifying earmarks in congressional bills. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 303–311). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/2939672.2939711.

Xu, X., Zhu, Y., Wang, X., & Zhang, N. (2023). How to unleash the power of large language models for few-shot relation extraction? In *SustaiNLP* (pp. 190–200). URL: https://doi.org/10.18653/v1/2023.sustainlp-1.13.

Xue, B., & Zou, L. (2023). Knowledge graph quality management: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, *35*(5), 4969–4988. http://dx.doi.org/10.1109/TKDE.2022.3150080.

Zahera, H. M., Elgendy, I. A., Jalota, R., Sherif, M. A., & Voorhees, E. (2019). Fine-tuned BERT model for multi-label tweets classification.. In *Proceedings of the twenty-eighth text rEtrieval conference, {tREC} 2019, gaithersburg, maryland, USA, November 13-15, 2019* (pp. 1–7). URL: https://trec.nist.gov/pubs/trec28/papers/DICE_UPB.IS.pdf.

Zhao, J., Wang, T., Abid, W., Angus, G., Garg, A., Kinnison, J., et al. (2024). LoRA land: 310 fine-tuned LLMs that rival GPT-4, a technical report. arXiv preprint arXiv:2405.00732.

Zhu, Y., Wang, X., Chen, J., Qiao, S., Ou, Y., Yao, Y., et al. (2024). LLMs for knowledge graph construction and reasoning: recent capabilities and future opportunities. *World Wide Web*, *27*(5), http://dx.doi.org/10.1007/s11280-024-01297-w.