JuriFindIT: an Italian legal retrieval dataset

Niko Dalla Noce[1]*, Davide Colla[2], Sina Farhang Doust[2], Lorenzo De Mattei[2], Davide Bacciu[1,2]

[1]University of Pisa - Computer Science Department, [2]Aptus.AI
niko.dallanoce@phd.unipi.it, {davide, sina, lorenzo}@aptus.ai, davide.bacciu@unipi.it

Abstract

Statutory article retrieval (SAR) targets retrieval of legislative provisions relevant to a natural language question. The lexical gap between everyday queries and specialized legal language, as well as the structural dependencies of statute law, makes it a challenging task. Here, we introduce JuriFindIT, the first SAR dataset for the Italian legal domain and the first to explicitly encode cross-article references extracted from national legal code. The dataset covers four macro-areas—civil law, criminal law, anti-money laundering and counter-terrorism, and privacy—and includes 895 expert-authored questions and 169,301 generated ones, linked to more than 23,000 statutory articles. We provide retrieval models fine-tuned on JuriFindIT, proposing a pipeline that integrates dense encoders with a heterogeneous legislative graph, achieving consistent improvements over prior SAR approaches.

1 Introduction

Access to justice increasingly depends on the ability of citizens and professionals to navigate complex bodies of statutory law. Yet, the language of statutes is highly technical, their organization fragmented across national and European sources, and existing search tools are not always well suited to address practical legal questions. This complexity makes it difficult to efficiently identify the provisions relevant to a given issue, creating both societal and technical challenges (Ponce et al., 2019; Balmer et al., 2010).

Research on legal information retrieval is a rapidly expanding field, with applications ranging from case law search (Chalkidis et al., 2019) to contract analysis (Hendrycks et al., 2021a; Tuggener et al., 2020). Within this landscape, statutory article retrieval (SAR) has emerged as a key element of the legal NLP pipeline (Shao et al., 2020). Unlike traditional ad-hoc retrieval (Craswell et al., 2020), SAR must bridge a pronounced lexical and semantic gap between everyday queries and the specialized legal formulations found in statutes. SAR challenges are multi-faceted. While individual articles are formally self-contained, their interpretation and practical application need considering surrounding provisions and citations to related documents. Regional aspects also play a major role that goes beyond language itself, but rather touch upon the nation-specific structuring of legal code and its interpretation practices. Current work in this area has modeled statutes primarily through their hierarchical organization. BSARD (Louis and Spanakis, 2022; Louis et al., 2023) is a relevant example of citizen-centered benchmark of statutory article retrieval in French, that maps jurist-annotated questions to Belgian law, but without incorporating cross-article references.

In this work, we present *JuriFindIT*, the first SAR benchmark devised specifically for the Italian legal domain. It is, to the best of our knowledge, also the first dataset to explicitly encode references between articles. As such, JuriFindIT enables richer structural modeling and broadens the linguistic and jurisdictional coverage of statutory retrieval. The dataset covers four core areas of legal practice: civil law, criminal law, anti-money laundering and counter-terrorism, and privacy. Our contributions are threefold. (i) Dataset: We release JuriFindIT[1], the first curated Italian statutory article retrieval dataset enriched with explicit cross-article references. (ii) Model: We fine-tune a strong retrieval model[2] on JuriFindIT to establish competitive baselines and to support the development of systems tailored to the Italian legal domain. (iii) Pipeline: We propose a revisited retrieval pipeline

*Work done during intership at Aptus.AI.

[1] JuriFindIT

[2] DAR-legal-it

Findings of the Association for Computational Linguistics: EACL 2026, pages 4223–4241
March 24-29, 2026 ©2026 Association for Computational Linguistics

that integrates dense encoders (Karpukhin et al., 2020) with a legislative graph, achieving improved performance over prior approaches to statutory article retrieval. We believe that JuriFindIT provides a fundamental resource for advancing research in the Italian legal domain and fostering reproducible evaluation. The dataset, the retrieval model, and code[3] will be publicly released to support future research.

2 Related works

Legal information retrieval spans across multiple tasks. The most established benchmark in case-law is COLIEE (Goebel et al., 2024), providing annual challenges where systems retrieve relevant precedents from Canadian or Japanese corpora. Larger-scale resources such as CAP (Harvard Law School Library Innovation Lab, 2018) offer millions of U.S. court opinions, enabling research under realistic search conditions. ECtHR-PCR (Santosh et al., 2024) provides a temporally-split benchmark for prior-case retrieval at the European Court of Human Rights. Work on similar-case matching frames retrieval as a constrained decision task. CAIL2019-SCM (Xiao et al., 2019) provides triplets of Chinese Supreme People's Court cases and challenges systems to identify the most analogous pair, reflecting how legal practitioners compare cases under fine-grained factual distinctions.

Contract-domain retrieval is supported by LEDGAR (Tuggener et al., 2020) (clause-level classification from SEC filings) and CUAD (Hendrycks et al., 2021b) (expert-annotated agreements), balancing scale with annotation quality. Statutory retrieval has similarly benefited from resources like EUR-Lex and MULTI-EURLEX (Chalkidis et al., 2021), which focus on multilingual legislative search.

BSARD (Louis and Spanakis, 2022) and LLeQA (Louis et al., 2024) expand research beyond professional settings, which is crucial for citizen-centered legal NLP. BSARD pairs 1,100 jurist-annotated legal questions from Belgian citizens with statutory references, creating a realistic benchmark for statute retrieval. LLeQA extends this setting to long-form legal question answering, requiring models to synthesize answers across multiple articles rather than retrieving a single provision.

LexGLUE (Chalkidis et al., 2022) aggregates datasets like LEDGAR into a unified evaluation suite, enabling standardized comparisons across retrieval and classification tasks. Adaptive re-ranking methods exploit corpus-level graph structure to refine first-stage retrieval by modeling document–document relationships (MacAvaney et al., 2022). Synthetic query generation has been shown to boost retrieval quality via generate-and-distill strategies for cross-language IR (Lawrie et al., 2025). Finally, CLERC provides a large-scale dataset for U.S. legal case retrieval and retrieval-augmented analysis generation, underscoring the importance of high-quality legal IR benchmarks for downstream reasoning (Hou et al., 2025).

3 JuriFindIT: an Italian legal retrieval dataset

We present JuriFindIT, a novel dataset for statutory article retrieval in the Italian legal domain. Figure 1 provides a high-level overview of the dataset creation process, which combines expert-written queries with systematic encoding of heterogeneous European and national legal sources. The resulting corpus explicitly models document hierarchies and cross-references, offering a high-quality benchmark for legal information retrieval.

3.1 Article corpus

The corpus was constructed by parsing files in the Akoma Ntoso format (Vitali et al., 2018), resulting in a heterogeneous collection of normative and para-normative sources adopted at both European and national levels. It comprises legislative acts such as regulations, directives, decisions, legislative decrees, and laws, alongside secondary instruments including provisions, circulars, guidelines, communications, and announcements issued by independent authorities. A full breakdown of the emitting entities and document distribution is reported in appendix A. In the remainder of this paper, we use the term *document* to denote a structured unit parsed from an Akoma Ntoso file, and the term *article* to refer to its textual content nodes, excluding the hierarchical structure.

Hierarchical structure. Statutory article retrieval is shaped by the hierarchical organization of legal codes, which guide experts from broad categories to specific provisions (Louis et al., 2023). Moreover, the meaning of a provision often depends on neighboring articles, making cross-article links essential for legal reasoning. We capture these dimensions by encoding the hierarchical path of

[3] JuriFindIT - GitHub

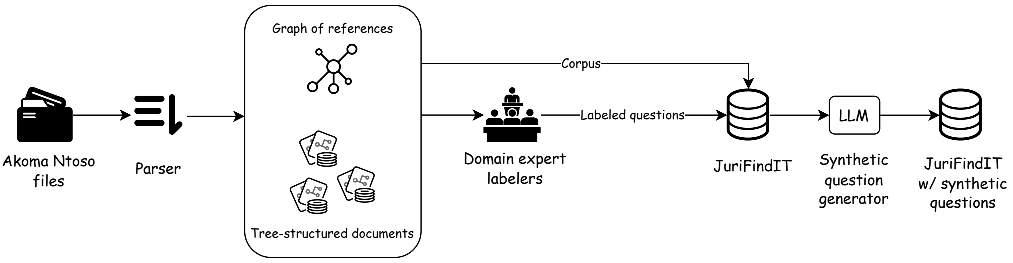

Figure 1: Overview of the parsing and annotation pipeline used in the *JuriFindIT* dataset creation process.

each record together with its source file and introducing synthetic root nodes (denoted as '*«file root node»*' with path '/') to represent the top level of each document. This design allows us to explicitly model references that target an entire document rather than a specific article, enabling their inclusion as edges in a legislative graph (details in § 3.2). Articles vary in length and structural consistency both across and within authorities. To improve corpus quality, we standardized records by applying ad-hoc parsing rules.

Article references. We leveraged a proprietary machine learning system developed by Aptus.AI[4] to extract internal and external references across legal articles, as such metadata is typically not released by the issuing authorities. The system builds upon the XLM-RoBERTa architecture (Conneau et al., 2020) and is fine-tuned using a BIO tagging scheme (Ramshaw and Marcus, 1995) to jointly perform reference detection, segmentation, and classification. Importantly, the model does not only identify spans corresponding to legal references, but also categorizes them by type (e.g. chapters, sections, and paragraphs). We applied this tool to every record to systematically extract all references occurring in the text, resolving each recognized reference to its corresponding article identifier. This pipeline enabled the automatic extraction of both explicit and otherwise implicit cross-references, resulting in a corpus with a richer representation of the structural and semantic interconnections among legal and regulatory provisions. Performance of the reference extraction system is provided in appendix C.

3.2 Legislative graph

We build a heterogeneous graph with two types of nodes: (i) *hierarchical nodes* representing structural elements such as article roots, sections, and chapters, and (ii) *content nodes* containing the actual article text. Formally, we define the legislative graph as $G = (V, E)$, where $V = V_{\text{hier}} \cup V_{\text{cont}}$ is the set of hierarchical and content nodes, and $E = E_{\text{struct}} \cup E_{\text{ref}}$ contains structural edges (linking roots, sections, chapters, and articles) and reference edges (capturing internal and external cross-references between articles). Each node of type $y \in \{\text{hier}, \text{cont}\}$ is initialized with a d-dimensional vector, yielding a feature matrix $X_y \in \mathbb{R}^{|V_y| \times d}$. Details on the initialization of X are provided in § 4.2. Then, we define the set of structural edges as $E_{\text{struct}} = \{(V_{\text{hier}}, V_{\text{cont}}), (V_{\text{cont}}, V_{\text{hier}}), (V_{\text{hier}}, V_{\text{hier}})\}$ and the set of reference edges as $E_{\text{ref}} = \{(V_{\text{cont}}, V_{\text{cont}}), (V_{\text{cont}}, V_{\text{hier}})\}$.

For each document, we parse the path associated with each article and create one hierarchical node per element separated by the delimiter "__". For example, a path like *chapter_2__article_11* results in two nodes, *chapter_2* and *article_11*; the former is connected to the article root, while the latter is linked to its corresponding article text. This procedure yields a directed acyclic graph for each file. We then add edges in E_{ref} for both internal and external references, connecting the corresponding nodes across files. Finally, we introduce a synthetic hierarchical super-root node and connect it to the roots of all files, resulting in a unified graph G.

3.3 Question labeling

In this work, we collaborated with the legal team of Aptus.AI that provides legal assistance services to professionals and enterprises. Dataset questions were authored directly by the legal team, composed of four experts holding law degrees—each specializing in one of four areas: criminal law, civil code, anti-money laundering and counter-terrorism regulations, and privacy.

The experts wrote questions representative of the most common issues encountered in profes-

[4] https://aptus.ai

sional legal contexts on their domain-specific experience. These questions were then used as the basis for the annotation process, in which the legal experts exhaustively identified and labeled all relevant articles that could provide an answer. Finally, each question is annotated with fine-grained topics that further refine the four macro-categories. Table 2 presents representative questions together with their associated topics, normative areas, and relevant statutory articles, while additional examples and discussion are provided in appendix A.1.

Annotation procedure. The annotation procedure was organized around the notion of *primary* and *secondary* norms. For each normative area, the team first manually selected a set of primary norms from the available documents, focusing on provisions containing the main sanctions, i.e., penalties incurred when an obligatory statute is violated. For each primary norm, annotators then derived a set of keywords summarizing its content. A keyword-based search system was used to retrieve other norms potentially related to that primary norm, which were grouped as secondary norms, thus yielding a hierarchical structure of primary–secondary links. Given this structure, annotators first selected the primary norms that could plausibly address a given question, discarding many non-relevant primary and secondary norms in the process. They then examined only the secondary norms associated with those selected primary norms, and finally annotated as relevant the subset of norms (primary and/or secondary) that contributed to answering the question.

On average, expert annotators required approximately 15 minutes per question to exhaustively identify all relevant articles. Assuming an hourly cost of €200, the resulting set of annotated queries corresponds to an estimated value of about €45,000.

Synthetic questions. We used a large language model to expand the set of expert-annotated queries. For each article, the model generated up to eight synthetic questions—depending on article length—closely mimicking the style of expert-written ones. We report the details in appendix A.2. A linguistic profiling of labeled and synthetic questions, based on Profiling–UD features (Brunato et al., 2020), is provided in appendix A.3 and shows that synthetic questions largely preserve the lexical and syntactic properties of expert-authored ones.

3.4 Dataset analysis

We present quantitative statistics on the article corpus, the inter-article connections, and the question–article annotations.

Corpus. Our final corpus comprises 23,458 articles, excluding synthetic root nodes, sourced from 159 Akoma Ntoso files spanning both European and national levels. On average, each article contains 207 words, with 99% containing fewer than 1,955 words. A total of 753 records contain fewer than five words, most of which correspond to structural elements such as chapter, section, or paragraph headings. At the other end of the distribution, 227 articles exceed 2,000 words; the majority of these are notes or attachments, while the remaining cases correspond to long-form articles lacking a defined internal structure that could be further segmented.

References. We extracted 20,608 references from the article texts, with 3,814 articles containing at least one reference to another document/article. Here, we will not make a distinction about their type as all the references are resolved to their corresponding article identifier, as described in § 3.1. The distribution of references per article is long-tailed, with 90% of articles containing at most seven references, and both the mode and median being equal to one. A reason behind this sparsity could be traced to the modular nature of legal drafting, where most provisions are designed to be self-contained (Goebel et al., 2024) and cross-references are introduced only when necessary to maintain coherence or avoid redundancy.

Question–article annotations. The annotation process described in § 3.3 produced 895 samples. Each sample consists of a question paired with one or more articles labeled as relevant. Questions are also grouped as outlined in § 3.3. We collected 144, 574, 108, and 69 questions for each category, having an average length of 21, 27, 34 and 22 words respectively. Table 1 shows substantial variation in the number of relevant articles per question across macro-areas. Civil code and criminal law questions are highly localized, with medians of 1 and 95% of questions requiring at most three relevant articles. In contrast, AML/CTF (Anti-Money Laundering / Counter-Terrorist Financing) and privacy questions exhibit a long-tailed relevance distribution, with median numbers of relevant articles per question of 4 and 15, respectively, and even their 50th

Macro-area	Mean	P(50)	P(75)	P(95)	Max
Criminal law	1.59	1.00	2.00	3.00	6
Civil code	1.03	1.00	1.00	1.00	4
AML & CTF	14.50	4.00	15.25	58.20	292
Privacy	23.45	15.00	30.00	96.00	113

Table 1: Statistics of the number of relevant articles per question, broken down by macro-area. P(k) indicates k*-th* percentile.

percentiles exceeding the 95th percentile of civil code and criminal law. This indicates that questions in these domains often require consulting a broader set of provisions, which increases retrieval difficulty and motivates the use of recall-oriented evaluation metrics.

Analyzing the dataset from another perspective, we observe a total of 2,439 distinct articles used to label the questions. Article overlap across questions reveals a median of one article per question and a 95th percentile of three. This contained overlap indicates that most questions are annotated with different articles. Such diversity increases the representativeness of the dataset and supports a more robust evaluation of retrieval systems, as models must generalize across a wide range of sources rather than overfitting to a small subset.

4 Article Retrieval

Given a legal question in natural language, the goal of a statutory article retriever is to return a small set of legislative articles that are relevant to answering the question. We model retrieval as a function

$$R : (q, C; k) \rightarrow F \tag{1}$$

that takes a query q and a corpus $C = \{a_1, \ldots, a_N\}$ of N articles, returning a ranked subset $F \subset C$ of size $k \ll N$. Let $s_\theta(q,a) \in \mathbb{R}$ be a scoring function parameterized by θ (e.g., a dense retriever), and let $\text{rank}_q(a)$ denote the rank of a when sorting C by decreasing $s_\theta(q,a)$. We define:

$$\begin{aligned} F = R(q, C; k) &= \{\, a \in C \;\mid\; \text{rank}_q(a) \leq k \,\} \\ &= \operatorname*{TopK}_{a \in C}\; s_\theta(q,a). \end{aligned} \tag{2}$$

Retrievers are typically evaluated with rank-based metrics such as Recall@k or MRR@k. Modern pipelines adopt a two-stage design (Guo et al., 2022; Tao et al., 2023), where a high-recall retriever first collects candidate articles F and a reranker refines their order. We focus on the first stage, which is critical to ensure that relevant articles are not missed. Building on the two-stage design proposed by Louis et al. (2023), we adopt a retriever that first uses a dense bi-encoder to map queries and candidate articles into a shared representation space. Unlike their architecture, which relies on two Transformer encoders plus a hierarchical aggregation layer, we employ a single Transformer encoder with an extended context window that directly encodes full articles, simplifying training, reducing parameters, and improving both efficiency and performance. Next, a graph neural network (GNN) (Bacciu et al., 2020) propagates relevance signals across a heterogeneous legislative graph comprising hierarchical and content nodes connected through multiple edge types, including explicit cross-article links, to refine article representations. An overview of these two components is provided in fig. 2, with further details in § 4.1 and § 4.2.

4.1 Dense article retriever

Transformer-based siamese bi-encoders (Reimers and Gurevych, 2019) are commonly used for dense passage retrieval (Karpukhin et al., 2020). They consist of two identical Transformer (Vaswani et al., 2017) encoders $\text{Enc}_\theta(\cdot)$ that map queries and articles into a shared d-dimensional representation space. Given a query q and an article a, we compute their vector representations as: $\mathbf{h}_q = \text{Enc}_\theta(q)$, $\mathbf{h}_a = \text{Enc}_\theta(a)$, $\mathbf{h}_q, \mathbf{h}_a \in \mathbb{R}^d$.

Sentence representations are obtained by applying mean-pooling over token embeddings, i.e. $\mathbf{h} = \frac{1}{T}\sum_{j=1}^{T} t_j$, where $t_j \in \mathbb{R}^d$ is the embedding of the j-th token and T is the sequence length.

The similarity between a query and an article is computed using cosine similarity. Formally, given a query q and an article a, we define

$$s_\theta(q,a) = \cos\left(\mathbf{h}_q, \mathbf{h}_a\right) = \frac{\mathbf{h}_q \cdot \mathbf{h}_a}{\|\mathbf{h}_q\|\,\|\mathbf{h}_a\|}, \tag{3}$$

where h_q and h_a denote the corresponding dense representations.

Given a corpus C, retrieval is then performed by ranking all articles $a \in C$ according to $s_\theta(q,a)$ and returning the top-k results (eq. (2)). Within our setting, we call this phase *DAR* (Dense Article Retrieval).

4.2 Graph encoding

To propagate structural context and improve DAR representations, our graph-informed retriever (GIR)

question	**topics**	**normative_area**	**relevant_doc_ids**
A citizen burns waste near a natural reserve; can this lead to criminal consequences?	Criminal law – environmental crimes, environmental pollution	Criminal Code	[17247, 17254]
A company remains with a single shareholder for more than six months. Can it be dissolved for this reason?	Civil law – grounds for dissolution of a company	Civil Code	[21549]
An individual receives a bank check made payable to the drawer and endorses it to a private party instead of depositing it at a bank...	AML – sanction for endorsing a check payable to the drawer	AML and CTF	[6425, 6428]

Table 2: Examples of annotated questions from the JuriFindIT dataset, translated into English. Each entry includes the question, its fine-grained topics, the broader normative area, and the list of relevant statutory articles ids.

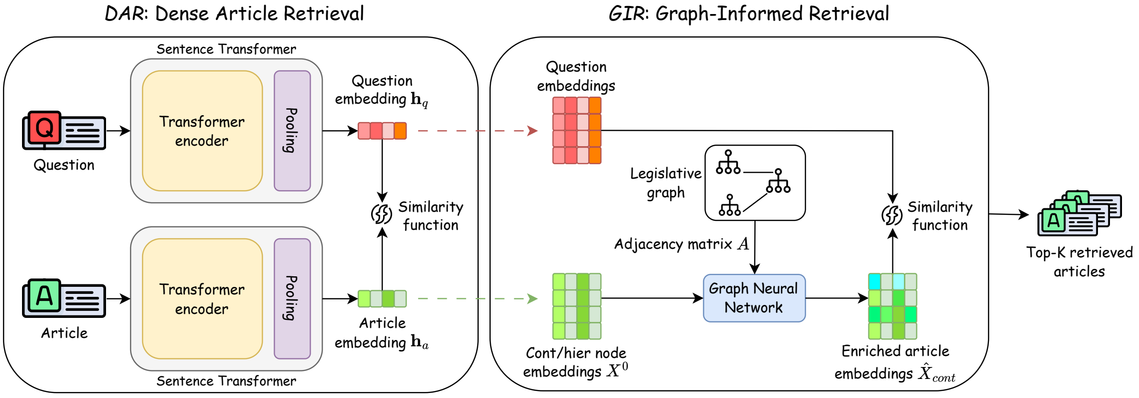

Figure 2: An overview of the two main components of our methodology. Left: *DAR* encodes questions and articles into a shared embedding space to align relevant pairs. Right: *GIR* enriches article embeddings with legislative graph information via a GNN, retrieving the top-k most relevant articles for each query.

applies a graph neural network on the legislative graph (§ 3.2).

A GNN is a stack of L message-passing layers: at each layer, every node aggregates information from its neighbors, updates its hidden state, and forwards it to the next layer. For each node type y, the feature matrix X_y is initialized with a d-dimensional vector from the bi-encoder retriever, forming the initial feature matrix $X_y^{(0)}$. Edges are typed according to their source node type (s) and target node type (t). For each edge type $e = (s, t)$, we define an adjacency matrix $A_e \in \{0, 1\}^{|V_t| \times |V_s|}$, where $A_e(v, u) = 1$ if there is an edge of type e from node $u \in V_s$ to node $v \in V_t$.

A GNN produces node representations $\hat{X}_y \in \mathbb{R}^{|V_y| \times d}$ that capture both node attributes and local graph structure. A general formulation of the ℓ-th GNN layer is as follows:

$$X_y^{(\ell+1)} = \sigma(X_y^{(\ell)}, A), \tag{4}$$

with $\hat{X}_y = X_y^{(L)}$, and $\sigma(\cdot)$ a nonlinear function. Each edge type e and layer ℓ has a learnable weight matrix $W_e^{(\ell)} \in \mathbb{R}^{d \times d}$, used to transform the features of source nodes into the space of the target nodes. At layer $\ell + 1$, the features of nodes of type y are updated by aggregating information from all their neighbors:

$$X_y^{(\ell+1)} = \sigma\left(\mathrm{Agg}_y\left(\sum_{e \in E'_y} A_e\, X_s^{(\ell)}\, W_e^{(\ell)}\right)\right), \tag{5}$$

where E'_y is the set of edge types incoming to y, i.e. $E'_y = \{(s, t) : t = y\}$. $X_s^{(\ell)}$ are the features of the corresponding source nodes at layer ℓ, and $\mathrm{Agg}_y(\cdot)$ is a permutation-invariant aggregation function.

After L rounds of message passing, the final embeddings are $X_y^{(L)}$. For retrieval, we keep only the content-node embeddings, denoted as $\hat{X}_{cont}$.

4.3 Learning

We use contrastive learning to optimize both the bi-encoder and the GNN in a two-stage process. In the first stage, we train the bi-encoder by optimizing the negative log-likelihood of the positive article against the negative ones. We define the query-anchor pair training dataset with N instances as $\mathcal{D} = \{(q_i, a_i^p)\}_N$. For each query q_i with a relevant article a_i^p, we sample negative articles using in-batch negatives technique (Karpukhin et al., 2020). Let $\mathcal{N}(q_i)$ be the set of negatives articles for a question q_i. For a batch of size B, it is defined as $\mathcal{N}(q_k) = \{a_j : k \neq j\}$ with $k, j \in [1, B]$. The contrastive loss is formulated as:

$$\mathcal{L}(q_i, a_i^p, \mathcal{N}(q_i)) = -\log \frac{e^{s_\theta(q_i, a_i^p)/\tau}}{\sum_{a \in \{a_i^p\} \cup \mathcal{N}(q_i)} e^{s_\theta(q_i, a)/\tau}}, \tag{6}$$

where $s_\theta(q_i, a)$ denotes the similarity between the query q_i and article a, and τ is the temperature parameter. In the second stage, we apply the same contrastive objective, this time using the article embeddings encoded in the matrix $\hat{X}_{\text{cont}}$ as produced by the GNN. The query embeddings are pre-computed offline by the trained bi-encoder and kept fixed during this stage. This step encourages the GNN-produced article representations to remain aligned with the semantic space of the bi-encoder, enabling the model to incorporate both textual and structural signals in the final retrieval stage.

5 Experiments

5.1 Experimental setup

DAR models. DAR sentence transformer was selected by looking at the MTEB benchmark ranking (Muennighoff et al., 2023). Due to hardware and time constraints, we did not simply select the top-performing models in the information retrieval category. Instead, we considered two practical factors: (i) the number of model parameters, which we limited to at most 600M, and (ii) a sufficiently large input context window, requiring support for more than 4096 tokens. Based on these criteria, we selected two pre-trained models for our experiments: *jina-embeddings-v3*, built on XLM-RoBERTa (Conneau et al., 2020), *snowflake-arctic-embed-m-v2.0*, based on GTE-multilingual (Zhang et al., 2024) and *Qwen3-Embedding-0.6B* the 0.6B variant of *Qwen3-Embedding* family of models (Zhang et al., 2025). These models support extended context lengths through rotary positional encoding (RoPE) (Su et al., 2024), making them well suited for handling long legal articles. Although pre-trained models specifically tailored to Italian legal text are scarce, we identified one suitable for this task: *Italian-Legal-BERT* (Licari and Comandè, 2024), a BERT-based model (Devlin et al., 2019) further pre-trained on Italian legal corpora. We cannot evaluate the pre-trained performance of this model as it does not come with a pooling layer. Despite its shorter input context window (limited to 512 tokens), this model enables us to assess the contribution of domain-specific pre-training to retrieval performance. Finally, given its excellent performance on MTEB, we also tested the *Qwen3-Embedding* variant with 8B of parameters.

GIR models. We compared several GNN architectures, including GraphConv (Kipf and Welling, 2017), GraphSAGE (Hamilton et al., 2017), GAT (Velickovic et al., 2018), GATv2 (Brody et al., 2022), and HGT (Hu et al., 2020). We further examined the impact of including references between articles. An ablation study (appendix E.4) showed that GATv2 performed best for the graph without references, while GAT was preferable when references were included. The architecture consists of an input layer matching the embedding dimension of the DAR model (768), followed by a single GATv2/GAT layer with hidden size 192 and one attention head, and an output layer projecting back to 768 to enable direct comparison between graph-informed embeddings and those produced by the question encoder.

Evaluation and metrics. We allocated 80% of the dataset to training and the remaining 20% to validation. After an ablation study (appendix E.1), we augmented the training split with one synthetic question per article, as described in § 3.3. Given that the validation set is composed of 179 samples, we further assessed the robustness of our results through a complementary 5-fold cross-validation analysis, reported in appendix D, which confirms that performance remains stable across folds. Retrieval performance was evaluated with standard metrics at cutoff k, considering only the top-k retrieved articles (appendix F). Recall@k measures the proportion of relevant items retrieved, mAP@k averages precision over relevant items, nDCG@k rewards ranking relevant items higher, and MRR@k reflects the rank of the first relevant hit. As many queries contain multiple relevant articles, Recall@k, mAP@k, and nDCG@k are most appropriate, as they capture both coverage and ranking quality. MRR@k is less informative in this setting but is reported for completeness, given the small number of relevant articles per query in the criminal law and civil code domains (§ 3.3). For

Recall we report $k \in \{5, 20, 60, 100\}$, covering the 95th percentile across macro-areas and the largest median number of articles per query (table 1). For the other metrics, since performance varies less with k, we report only the two boundary values.

Training Details. We trained Transformer models with AdamW (Loshchilov and Hutter, 2019), using a maximum learning rate of 3.5×10^{-5}, 5 warm-up steps, and a linear scheduler. An ablation study set the batch size to 768 (appendix E.2), made feasible by GradCache (Gao et al., 2021). We truncate texts exceeding model's maximum context length. We trained the GIR module with SGD at a learning rate of 3.5×10^{-3}, momentum 0.3, dropout 0.15, and batch size 4096. To stabilize and speed-up training, we used GraphNorm (Cai et al., 2021) and residual connections. Model selection relied on mean average precision (mAP), which improved more consistently than recall, with early stopping applied on mAP using a patience of 5 epochs.

Hardware and libraries. All models were implemented and trained on a single NVIDIA A100 GPU with 80GB of memory, paired with an AMD EPYC 7413 24-core CPU. The software stack included *PyTorch* (Paszke et al., 2019) and *PyTorch Lightning*[5] for model development and training, *PyTorch Geometric* (Fey and Lenssen, 2019) for GNN implementations, and the *HuggingFace Transformers* (Wolf et al., 2020), *Sentence-Transformers*[6], and *Datasets* (Lhoest et al., 2021) libraries for Transformer models and data handling.

5.2 Results

Table 3 reports retrieval performance across all model variants. We observe that all pre-trained embedding models consistently outperform the BM25 baseline across both recall and rank-aware metrics, underscoring the advantage of contextualized representations over lexical matching.

Fine-tuning consistently boosts retrieval quality. Among the evaluated models, *snowflake-arctic-embed-m-v2.0* achieves the strongest overall results, both in terms of average performance across metrics and Recall@5. The latter is particularly important, as most queries are associated with fewer than five relevant articles. For this reason, we adopt *snowflake-arctic-embed-m-v2.0* as our DAR backbone. Results from *Qwen3-Embedding-8B* further highlight that fine-tuning a smaller model remains advantageous, although its zero-shot performance already approaches that of our best fine-tuned system and even surpasses the fine-tuned *Italian-Legal-BERT*. In the absence of an explicit reranker, rank-aware metrics such as nDCG, mAP, and MRR remain essential for assessing retrieval quality, whereas with a reranking stage, recall naturally becomes the primary optimization target.

In the DAR+GIR setting, we assess the impact of incorporating a legislative graph. The variant labeled *legislative graph w/o references* in table 3, which excludes explicit references between articles, achieves higher recall at lower cutoffs ($k = 5, 20$). Conversely, the variant labeled *legislative graph w/ references*, which incorporates such references, performs slightly better at larger k values, while rank-aware metrics remain broadly comparable. Notably, since GIR yields small but consistent gains also on rank-sensitive measures, it slightly improves re-ranking on top of the DAR representations. Averaging across metrics, both graph-enhanced variants outperform the pure DAR model, showing that graph structure provides complementary signal for retrieval.

From a modeling perspective, these gains reflect the relational nature of statutory law: articles form a network of hierarchical dependencies and explicit cross-references that guide their interpretation. By propagating relevance signals along these edges, GIR can surface provisions that are not lexically closest to the query but are interpretively linked to highly ranked articles, complementing the semantic matching performed by DAR. This motivation is consistent with prior graph-based approaches to statutory retrieval (Louis et al., 2023), and the fact that improvements are most pronounced in rank-aware metrics such as nDCG and mAP suggests that the graph mainly refines the ordering among plausible candidates rather than simply increasing recall.

5.3 Benchmark comparison

To contextualize our benchmark and assess the generality of the results, we also evaluate the best-performing methods on BSARD (table 4) and LLeQA (table 5). On BSARD, we improved first-stage retrieval by generating five additional synthetic questions and incorporating a heterogeneous graph, processed with a one-layer SAGE GNN (hidden size 2304) on top of the same fine-tuned transformer used in our main experiments.

[5] https://lightning.ai/

[6] https://sbert.net/

Model	#Params	Recall (↑)				nDCG (↑)		MRR (↑)		mAP (↑)	
		@5	@20	@60	@100	@5	@100	@5	@100	@5	@100
Baseline											
BM25	–	43.02	59.07	67.55	70.76	38.02	44.85	38.00	40.33	34.47	35.76
Pre-trained											
jina-embeddings-v3	572M	58.41	66.05	73.67	74.89	51.78	55.96	51.67	53.31	48.43	48.91
snowflake-arctic-embed-m-v2.0	305M	63.77	71.97	76.31	79.89	59.31	63.03	60.88	62.09	55.00	55.50
Qwen3-Embedding-0.6B	596M	58.82	70.74	77.65	79.52	54.27	59.04	56.13	57.92	49.84	50.51
Qwen3-Embedding-8B	7.57B	74.91	81.44	84.95	86.74	72.46	74.43	75.23	76.09	67.98	67.25
DAR (fine-tuning)											
jina-embeddings-v3	572M	74.43	**85.94**	**90.05**	**92.17**	71.67	76.20	75.47	76.60	66.96	67.81
snowflake-arctic-embed-m-v2.0	305M	76.02	83.31	88.61	91.29	74.19	77.59	77.20	78.01	69.85	70.15
Italian-Legal-BERT	111M	62.98	75.49	81.84	85.36	61.05	66.12	62.86	64.64	57.09	57.47
Qwen3-Embedding-0.6B	596M	73.96	83.99	89.42	90.56	70.72	74.57	72.81	74.08	65.82	65.89
DAR + GIR											
legislative graph w/o references	305M + 0.6M	**77.80**	83.57	87.75	90.55	**75.53**	77.96	77.84	78.34	**71.14**	**71.00**
legislative graph w/ references	305M + 0.5M	77.12	82.93	88.15	90.91	75.21	**78.00**	**78.08**	**78.67**	70.90	70.90

Table 3: Retrieval performance on the JuriFindIT validation set. Results are reported for the best-performing model run. Bold indicates the overall best score, and underlining highlights the top result within each setting (Baseline, Pre-trained, DAR, DAR+GIR). Higher is better.

This setting increased the number of parameters but consistently enhanced retrieval quality. On LLeQA, where neither synthetic data nor second-stage retrieval had been applied, our approach again yielded strong gains. Notably, recall@5 and recall@10 on JuriFindIT are higher than those on LLeQA, with our recall@5 even surpassing the recall@10 reported for LLeQA, while MRR@10 remains broadly comparable. We attribute this to the smaller number of labeled relevant articles per query in JuriFindIT, which inflates recall at low cutoffs. Larger input contexts and batch sizes further confirmed the effectiveness of our approach. Finally, while adding a second-stage retrieval module produced consistent gains, the effect was less pronounced in our setting, likely because our first-stage model (DAR) already outperforms the reported second-stage G-DSR baseline on BSARD. Metrics follow prior work to ensure comparability.

Model	#Params	Recall (↑)		mAP (↑)
		@100	@200	-
First stage				
DSR	234M	82.7	88.7	35.3
DAR (Ours)	305M	88.4	**91.9**	59.7
Second stage				
G-DSR	234M+28M	84.3	90.4	47.1
DAR+GIR (Ours)	305M+57M	**88.7**	91.7	**60.9**

Table 4: Retrieval performance on the BSARD test set. Bold marks the best score. '#Params' indicates the number of retriever / retriever + GNN parameters.

Model	#Params	Recall (↑)		MRR (↑)
		@5	@10	@10
First stage				
CamemBERT	111M	48.6	60.6	60.0
DAR (Ours)	305M	59.2	**71.4**	72.1
Second stage				
DAR+GIR (Ours)	305M+57M	**59.6**	70.2	**73.6**

Table 5: Retrieval performance on the LLeQA validation set. Bold marks the best score. '#Params' indicates the number of retriever / retriever + GNN parameters.

6 Conclusion

In this paper, we presented JuriFindIT, a new dataset for statutory article retrieval annotated by domain experts. The dataset provides a resource for evaluating and advancing retrieval models in the legal domain. We also validated a new retrieval pipeline, achieving performance that surpasses existing approaches on the two most comparable datasets. As future work, we plan to extend the dataset by including answers to the questions and by providing more fine-grained annotations at the paragraph level rather than the article level. We hope that this contribution will foster interest in developing practical and reliable models for legal article retrieval, ultimately supporting faster and more effective access to the relevant statutory sources needed to answer legal queries.

Limitations

While JuriFindIT provides a valuable resource for statutory article retrieval, it also presents several limitations. First, all questions and their relevant documents were annotated by a single team of experts from the same company. It is possible that another group of annotators would have associated different articles with each question, and relevant articles were not cross-verified by independent experts. A double-blind annotation process could have increased labeling reliability and overall dataset quality.

Second, the annotated questions cover only four macro-areas—civil code, criminal code, anti-money laundering and counter-terrorism regulations, and privacy. While representative of important domains, a broader coverage of additional legal areas would have allowed for more consistent and generalizable evaluation.

Third, access to real user-generated legal questions is limited. Companies that collect such data from citizens are often reluctant to share them without financial incentives, which slows progress in this research direction.

Fourth, the heterogeneity of the source files, stemming from different issuing authorities, makes parsing challenging; the resulting document structures are not always perfect and may contain errors. Similarly, inter-article references can currently only be extracted using a language model, which introduces potential inaccuracies and lacks standardized benchmarks for evaluation, limiting quality assurance to internal testing.

Fifth, our methodology enhances classic text retrieval by incorporating graph structure, but this approach is applicable only when documents can be modeled as a graph. In domains lacking such structural information, our method cannot provide the same benefits.

Sixth, the generation of synthetic questions remains an open challenge. As shown in appendix E.1, increasing the number of synthetic queries can actually reduce retrieval performance. This effect may stem from stylistic differences between generated and expert-authored questions, or from a bias introduced by having only a single relevant article per generated query. Future work could explore alternative strategies for query generation to mitigate these issues.

Finally, our pipeline does not yet include a reranking module, which could improve the quality of the top-k retrieved articles. Moreover, varying the value of k across domains—depending on the typical number of relevant articles per query—may further enhance retrieval performance. These improvements would be particularly beneficial if an LLM were integrated in the pipeline, as it would receive a more appropriately sized and ranked set of candidate articles to generate higher-quality answers.

Ethical considerations

The dataset is composed exclusively of statutory texts from national and European sources that are publicly available and free from copyright or confidentiality concerns. All questions were authored by domain experts, ensuring that no sensitive or personal user data are included. Nevertheless, parsing errors or automatic extraction of references may introduce inaccuracies, and we advise caution when applying the resource to downstream tasks. Moreover, while our retrieval methodology can facilitate access to legal information, it is not intended to replace professional legal advice. Responsible use is therefore essential, especially in scenarios where retrieval errors could affect decision-making processes.

Acknowledgments

This work has been partially supported by the European Commission under the NextGeneration EU programme – National Recovery and Resilience Plan project “FAIR-Future Artificial Intelligence Research” - Spoke 1 “Human-centered AI” (grant n. PE00000013).

References

Davide Bacciu, Federico Errica, Alessio Micheli, and Marco Podda. 2020. A gentle introduction to deep learning for graphs. *Neural Networks*, 129:203–221.

Nigel J. Balmer, Ash Patel, Alexy Buck, Catrina Denvir, and Pascoe Pleasence. 2010. *Knowledge, Capacity and the Experience of Rights Problems*. Public Legal Education Network: PLENet.

Shaked Brody, Uri Alon, and Eran Yahav. 2022. How attentive are graph attention networks? In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Dominique Brunato, Andrea Cimino, Felice Dell'Orletta, Giulia Venturi, and Simonetta Montemagni. 2020. Profiling-ud: a tool for linguistic profiling of texts. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 7145–7151. European Language Resources Association.

Tianle Cai, Shengjie Luo, Keyulu Xu, Di He, Tie-Yan Liu, and Liwei Wang. 2021. Graphnorm: A principled approach to accelerating graph neural network training. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 1204–1215. PMLR.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in english. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4317–4323. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. 2021. Multieurlex - A multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6974–6996. Association for Computational Linguistics.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael J. Bommarito II, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2022. Lexglue: A benchmark dataset for legal language understanding in english. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4310–4330. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *CoRR*, abs/2003.07820.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with pytorch geometric. *CoRR*, abs/1903.02428.

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021. Scaling deep contrastive learning batch size under memory limited setup. In *Proceedings of the 6th Workshop on Representation Learning for NLP, RepL4NLP@ACL-IJCNLP 2021, Online, August 6, 2021*, pages 316–321. Association for Computational Linguistics.

Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2024. Overview and discussion of the competition on legal information, extraction/entailment (COLIEE) 2023. *Rev. Socionetwork Strateg.*, 18(1):27–47.

Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Trans. Inf. Syst.*, 40(4):66:1–66:42.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034.

Harvard Law School Library Innovation Lab. 2018. Caselaw access project. `https://case.law`.

Matthew L. Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *CoRR*, abs/1705.00652.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021a. CUAD: an expert-annotated NLP dataset for legal contract review. In *Proceedings* *of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021b. CUAD: an expert-annotated NLP dataset for legal contract review. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Abe Bohan Hou, Orion Weller, Guanghui Qin, Eugene Yang, Dawn J. Lawrie, Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. 2025. CLERC: A dataset for u. s. legal case retrieval and retrieval-augmented analysis generation. In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 7898–7913. Association for Computational Linguistics.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2704–2710. ACM / IW3C2.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Dawn J. Lawrie, Efsun Kayi, Eugene Yang, James Mayfield, Douglas W. Oard, and Scott Miller. 2025. Generate-distill: Training cross-language IR models with synthetically-generated data. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*, pages 2926–2930. ACM.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Sasko, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, and 13 others. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2021, Online and Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 175–184. Association for Computational Linguistics.

Daniele Licari and Giovanni Comandè. 2024. ITALIAN-LEGAL-BERT models for improving natural language processing tasks in the italian legal domain. *Comput. Law Secur. Rev.*, 52:105908.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Antoine Louis and Gerasimos Spanakis. 2022. A statutory article retrieval dataset in french. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6789–6803. Association for Computational Linguistics.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2023. Finding the law: Enhancing statutory article retrieval via graph neural networks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 2753–2768. Association for Computational Linguistics.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2024. Interpretable long-form legal question answering with retrieval-augmented large language models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 22266–22275. AAAI Press.

Sean MacAvaney, Nicola Tonellotto, and Craig Macdonald. 2022. Adaptive re-ranking with a corpus graph. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 1491–1500. ACM.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. MTEB: massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 2006–2029. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan T. McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Alejandro Ponce, Sarah Chamness Long, Elizabeth Andersen, Camilo Gutiérrez Patiño, Matthew Harman, Jorge A. Morales, Ted Piccone, Natalia Rodríguez Cajamarca, Adriana Stephan, Kirssy González, Jennifer VanRiper, Alicia Evangelides, Rachel Martin, Priya Khosla, Lindsey Bock, Erin Campbell, Emily Gray, Amy Gryskiewicz, Ayyub Ibrahim, and 3 others. 2019. Global insights on access to justice 2019: Findings from the world justice project general population poll in 101 countries. https://worldjusticeproject.org/our-work/research-and-data/global-insights-access-justice-2019.

Lance A. Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora, VLC@ACL 1995, Cambridge, Massachusetts, USA, June 30, 1995*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

T. Y. S. S. Santosh, Rashid Haddad, and Matthias Grabmair. 2024. Ecthr-pcr: A dataset for precedent understanding and prior case retrieval in the european court of human rights. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/-COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 5473–5483. ELRA and ICCL.

Hsuan-Lei Shao, Yi-Chia Chen, and Sieh-Chuen Huang. 2020. Bert-based ensemble model for statute law retrieval and legal information entailment. In *New Frontiers in Artificial Intelligence - JSAI-isAI 2020 Workshops, JURISIN, LENLS 2020 Workshops, Virtual Event, November 15-17, 2020, Revised Selected Papers*, volume 12758 of *Lecture Notes in Computer Science*, pages 226–239. Springer.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Chongyang Tao, Jiazhan Feng, Tao Shen, Chang Liu, Juntao Li, Xiubo Geng, and Daxin Jiang. 2023. CORE: cooperative training of retriever-reranker for effective dialogue response selection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 3102–3114. Association for Computational Linguistics.

Don Tuggener, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. 2020. LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 1235–1241. European Language Resources Association.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Fabio Vitali, Monica Palmirani, Roger Sperberg, and Véronique Parisse. 2018. Akoma ntoso version 1.0. part 2: Specifications. OASIS Standard. Edited by Fabio Vitali, Monica Palmirani, Roger Sperberg, and Véronique Parisse.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Tianyang Zhang, Xianpei Han, Zhen Hu, Heng Wang, and Jianfeng Xu. 2019. CAIL2019-SCM: A dataset of similar case matching in legal domain. *CoRR*, abs/1911.08962.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: EMNLP 2024 - Industry Track,* *Miami, Florida, USA, November 12-16, 2024*, pages 1393–1412. Association for Computational Linguistics.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *CoRR*, abs/2506.05176.

A Dataset

The corpus comprises normative and para-normative sources issued by both European and national authorities. At the European level, the documents include acts from the Council of the European Union, the European Parliament acting jointly with the Council, and the European Commission with its directorates-general for Justice and Consumers (DG JUST) and for Financial Stability, Financial Services and Capital Markets Union (DG FISMA). Regulatory and supervisory bodies are also represented, such as the European Data Protection Board (EDPB), the European Banking Authority (EBA), the European Central Bank (ECB), and joint issuances of the European Banking Authority and the European Securities and Markets Authority (EBA–ESMA). The collection further includes treaties adopted by the Governments of the Member States (GOVREP) and guidance from the Article 29 Working Party (WP29), predecessor of the EDPB.

At the national level, the Italian State (Parliament and Government) accounts for the majority of documents, complemented by acts from the Ministry of Economy and Finance (MEF), the Ministry of Justice, the Ministry for Enterprises and Made in Italy (MIMIT), and the Presidency of the Council of Ministers through its Department for Digital Transformation (Dipartimento per la Trasformazione Digitale, PCM). Independent authorities also play a central role, including the Bank of Italy, the Financial Intelligence Unit (Unità di Informazione Finanziaria, UIF), the Italian Data Protection Authority (Garante per la protezione dei dati personali), the Insurance Supervisory Authority (Istituto per la Vigilanza sulle Assicurazioni, IVASS), and the National Anti-Corruption Authority (Autorità Nazionale Anticorruzione, ANAC). Additional contributions come from joint provisions by the Bank of Italy and UIF, collaborative issuances with MEF, the Agency for Digital Italy (Agenzia per l'Italia Digitale, AgID), and the Agents and Credit Brokers Body (Organismo Agenti e Mediatori, OAM). The distribution of documents across these entities is reported in table 6.

Entity	Count
Italian State (Parliament/Government)	35
European Parliament & Council (co-legislators)	23
Council of the European Union	20
UIF – Unità di Informazione Finanziaria	14
Bank of Italy	13
Italian Data Protection Authority (Garante)	11
European Data Protection Board	10
MEF – Ministry of Economy & Finance (Italy)	5
European Banking Authority	4
AgID – Agency for Digital Italy	3
Article 29 Working Party	3
European Commission – DG JUST	2
IVASS – Insurance Supervisory Authority (Italy)	2
Bank of Italy & UIF (joint)	2
ANAC – National Anti-Corruption Authority (Italy)	2
Bank of Italy, UIF & MEF (joint)	1
European Commission	1
European Commission – DG FISMA	1
Dip. Trasformazione Digitale – PCM (Italy)	1
Joint EBA & ESMA	1
European Central Bank	1
Governments of the Member States (EU Treaties)	1
Ministry for Enterprises and Made in Italy (MIMIT)	1
Ministry of Justice (Italy)	1
OAM – Agents and Credit Brokers Body (Italy)	1

Table 6: Distribution of documents across European and Italian emitting entities.

A.1 Dataset samples

Corpus. Table 7 provides an excerpt from the dataset, illustrating the main fields associated with each statutory article. The column *file_name* identifies the source file from which the article has been extracted, while *content* contains the textual content of the provision itself. Since documents are structured hierarchically, the *path* column specifies the location of the article within the tree rooted at the source file. Each article is assigned a unique identifier reported in the *id* column. Finally, the *reference* column lists the identifiers of other articles cited within the current one; for instance, the article with *id* 166 explicitly refers to articles 164 and 165.

Question-article labeling. Table 8 shows some samples of the annotated questions from the dataset. The *question* column reports the natural language query as formulated by the legal experts. Each question is associated with one or more *topics*, which provide a fine-grained categorization of the legal issues involved, while the *normative_area* column specifies the broader macro-area of reference (e.g.,

file_name	**content**	**path**	**id**	**reference**
§akn§eu§act§reg§CONSIL§2013-05-02§0401§ITA	Articolo 3 quater 1. A meno che l'autorità compete...	chapter_1__article_3-quater	163	[162 160 163]
§akn§eu§act§reg§CONSIL§2013-05-02§0401§ITA	Articolo 4 1. In deroga all'articolo 2, paragrafo 1, al...	chapter_1__article_4	164	[159 160 161 176 165]
§akn§eu§act§reg§CONSIL§2013-05-02§0401§ITA	Articolo 4 bis 1. Sono congelati tutti i fondi e le ris...	chapter_1__article_4-bis	165	[187]
§akn§eu§act§reg§CONSIL§2013-05-02§0401§ITA	Articolo 4 ter 1. In deroga all'articolo 4 bis, le auto...	chapter_1__article_4-ter	166	[165 164]

Table 7: Example rows from the dataset. Each entry reports the source file of the article (*file_name*), its textual content (*content*), the hierarchical path within the source document (*path*), a unique identifier (*id*), and references to other articles cited in the text (*reference*).

civil law, criminal law, privacy). The topics assigned by the annotators may be multiple for the same question, and care was taken to minimize the number of questions sharing the exact same set of topics, in order to ensure broader coverage of the normative areas. Finally, the *relevant_doc_ids* field lists the identifiers of the statutory articles deemed relevant for answering each question.

A.2 Synthetic question generation

To augment the dataset with additional training queries, we employed the Qwen3-32B[7] model (in non-thinking mode) to generate synthetic questions. Using the prompt reported below, the generation process took approximately 50 hours. The procedure was applied to each article in the corpus: for every article text, the LLM was instructed to generate exactly eight questions, guided by a few example queries randomly sampled from the annotated dataset. Articles whose length exceeded the available GPU memory were skipped. The expected output was a string containing a JSON object, from which we extracted the list of generated questions. The prompt used to generate synthetic questions is the following:

```
Sei un esperto legale. Dato il seguente
    testo legale sulla legge italiana,
    scrivi esattamente 8 domande che
    siano inerenti al contenuto del
    testo fornito.
Le domande devono essere analitiche e
    precise, basate sul testo
    legislativo fornito. Ogni domanda
    deve:
1. Essere focalizzata su dettagli
    concreti (es. definizioni, obblighi,
     condizioni, eccezioni, procedure,
    soggetti coinvolti), senza chiedere
    riassunti o spiegazioni generali.
2. Evitare riferimenti a numeri di
    articoli, comma o paragrafi (non
    usare espressioni come 'Nell'
    articolo X...').
3. Non essere generica (niente 'Di cosa
    parla il documento?' o 'Cosa
    stabilisce la sezione Y?').

Scrivi ogni domanda in un file json, del
     tipo {{ 'domande': [domanda1,
    domanda2,...] }}, dove la chiave e'
    'domande' ed il contenuto delle
    domande e' in una lista. Non
    aggiungere nient'altro nel json.

Basandoti esclusivamente sul contenuto
    sostanziale del testo, genera
    domande simili alle seguenti: '{1}'.

Ecco il testo dell'articolo:
'{0}'
```

A.3 Linguistic Analysis of Real vs. Synthetic Questions

To assess the linguistic similarity between expert-authored and synthetic questions, we conducted a detailed analysis using the *Profiling–UD* toolkit (Brunato et al., 2020). Profiling–UD is a multilingual system for linguistic profiling based on Universal Dependencies (Nivre et al., 2016), providing more than 130 features that describe surface, lexical, morphosyntactic, and syntactic properties of texts.

From the entire feature set, we selected a subset of metrics particularly suited for evaluating the stylistic and structural realism of automatically generated questions. At the surface and lexical level, we consider *n_tokens*, the total number of tokens in the question (a proxy for sentence length and verbosity); *char_per_tok*, the average number of characters per token (capturing lexical complexity and typical word length); and *lexical_density*, the proportion of content words (nouns, verbs, adjectives, adverbs) over total words, which measures informational load.

At the morphosyntactic level, we use the distributions *upos_dist_VERB* and *upos_dist_NOUN*, i.e., the relative frequencies of verbs and nouns according to the UD POS tagset, indicating how verbal or nominal a sentence is.

At the syntactic-structure level, we focus on

[7] https://huggingface.co/Qwen/Qwen3-32B

question	**topics**	**normative_area**	**relevant_doc_ids**
I genitori devono decidere insieme sull'istruzione e su...	Diritto civile – Esercizio congiunto della ...	Codice civile	[19243]
I sindaci di società quotate che espongono in modo non ...	Diritto penale – Disposizioni penali in mate...	Diritto Penale	[22123]
Luca e Giulia stipulano un contratto di compravendita i...	Diritto civile – Determinazione del prezzo ...	Codice civile	[20608]
Una banca utilizza un algoritmo per decidere automatica...	Privacy – Processo decisionale automatizzat...	Privacy	[1739, 1745, 1763, ...]
Marco, che possedeva un terreno credendo di esserne pro...	Diritto civile – Spese e miglioramenti del ...	Codice civile	[20223, 20224]

Table 8: The table presents sample annotated questions from the dataset. Each entry includes the query (*question*), its fine-grained *topics*, the broader *normative_area*, and the list of relevant statutory articles (*relevant_doc_ids*).

avg_max_depth, the mean maximum syntactic tree depth across sentences (capturing overall syntactic complexity as the longest path from the root to a leaf in the dependency tree); *avg_links_len*, the average dependency-link length, i.e., the linear distance in tokens between a syntactic head and its dependent (reflecting structural compactness); *avg_subordinate_chain_len*, the average depth of subordinate-clause embedding (measuring how many subordinate clauses are recursively embedded); and *subordinate_proposition_dist*, the proportion of subordinate clauses relative to all clauses (quantifying the overall reliance on subordination). Table 9 reports the mean and standard deviation for each feature across real and synthetic questions.

Analisys. The analysis highlights several convergences and differences between the two question sets. Real questions are longer on average (*n_tokens*) and slightly more lexically dense (*lexical_density*), whereas *char_per_tok* is almost identical, indicating comparable lexical sophistication. Synthetic questions contain fewer verbs and slightly more nouns (*upos_dist_VERB* vs. *upos_dist_NOUN*), suggesting a more nominal, declarative style. Both sets exhibit similar maximum syntactic depth (*avg_max_depth*), but synthetic questions show shorter dependency links (*avg_links_len*) and shallower subordinate chains (*avg_subordinate_chain_len*, *subordinate_proposition_dist*), reflecting more compact syntax and reduced reliance on subordination. Overall, these distributions indicate that synthetic questions retain key structural and lexical properties of expert-authored ones, while exhibiting moderate simplifications in length and subordination that are typical of automatically generated text.

B Legislative Graph Statistics

To better characterize the structure of the legislative corpus, we report a set of descriptive statistics over the graph defined in § 3.2. These statistics quantify the size of the graph in terms of nodes and edges, as well as structural properties of the trees induced by each document. A summary of all values is provided in table 10, while the formal definitions of the reported metrics are detailed below.

Graph statistics. Let $\mathcal{T}$ denote the set of trees induced by each file, with root r_t and leaves $\mathcal{L}(t)$. The **average height** of a tree defined as

$$\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \text{height}(r_t),$$

while the **average path length** in a tree formulated as:

$$\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \frac{1}{|\mathcal{L}(t)|}\sum_{\ell\in\mathcal{L}(t)} d(r_t,\ell),$$

where $d(r_t, \ell)$ is the length of the path from the root node to leaf ℓ.

C Reference extractor tool

As described in § 3.1, cross-article links were extracted using a proprietary reference extraction system. The company reports the following performance of the tool based on a two-stage evaluation: (i) Document-level identification: for a reference such as "Regolamento UE 2016/679, Articolo 10", the model is considered correct if it identifies the correct document (Regolamento UE 2016/679).
(ii) Section-level identification: the model is considered correct if it also detects the specific section within the document (here, Articolo 10).

On a manually verified set of 1,530 document references and 805 section references from EUR-Lex and Bank of Italy documents, the system achieved Precision = 92.21, Recall = 65.65, F1 = 76.84 at the document level, and Precision = 77.14, Recall = 52.29, F1 = 62.23 at the section level. The model is intentionally optimized for high precision, as incorrect references would inject noise into the legislative graph, whereas missing a small fraction of links has limited impact on retrieval.

Feature	**Description (short)**	**Real (mean ± std)**	**Synthetic (mean ± std)**
n_tokens	# tokens (length)	34.265 ± 12.002	24.816 ± 6.414
char_per_tok	avg chars per token	5.003 ± 0.511	5.029 ± 0.623
lexical_density	content-word ratio	0.501 ± 0.059	0.452 ± 0.061
upos_dist_VERB	% verbs	10.850 ± 3.986	6.660 ± 3.814
upos_dist_NOUN	% nouns	20.344 ± 4.598	22.937 ± 4.433
avg_max_depth	max tree depth (avg)	5.455 ± 1.626	5.676 ± 1.754
avg_links_len	dependency length (avg)	2.915 ± 0.658	2.214 ± 0.413
avg_subordinate_chain_len	subordinate chain depth (avg)	1.122 ± 0.630	0.815 ± 0.612
subordinate_proposition_dist	% subordinate clauses	57.779 ± 25.449	40.812 ± 27.787

Table 9: Linguistic statistics for real and synthetic questions computed with Profiling–UD. Values are reported as mean ± standard deviation.

Statistic	Value
Content nodes ($|V_{\text{cont}}|$)	23,458
Hierarchical nodes ($|V_{\text{hier}}|$)	48,911
Total nodes ($|V|$)	72,369
Structural edges ($|E_{\text{struct}}|$)	72,368
Reference edges ($|E_{\text{ref}}|$)	20,608
Total edges ($|E|$)	92,976
Average height	2.88
Average path length	3.38

Table 10: Overview of legislative graph statistics.

Metric	**k**	**Mean ± Std.**
Recall	5	0.7551 ± 0.0237
Recall	20	0.8521 ± 0.0251
Recall	60	0.8999 ± 0.0172
Recall	100	0.9212 ± 0.0130
nDCG	5	0.7276 ± 0.0143
nDCG	100	0.7571 ± 0.0264
MRR	5	0.7483 ± 0.0196
MRR	100	0.7579 ± 0.0195
mAP	5	0.6824 ± 0.0121
mAP	100	0.6842 ± 0.0142

Table 11: Five-fold cross-validation results on the JuriFindIT dataset using *snowflake-arctic-embed-m-v2.0* as the DAR backbone.

D Cross-Validation Analysis

To assess the robustness of our results beyond the single 80/20 split used in the main experiments, we conducted a 5-fold cross-validation on the expert-authored portion of the JuriFindIT dataset. Using *snowflake-arctic-embed-m-v2.0* as the DAR encoder and the same hyperparameters described in § 5.1, we observed stable performance across folds (table 11). These results indicate limited variance across folds and confirm that the model's performance is consistent despite the relatively small number of expert-authored queries in the validation set.

E Ablations

E.1 Synthetic questions

Table 12 presents the impact of varying the number of synthetic questions per article added to the original dataset. The ablation was conducted using *snowflake-arctic-embed-m-v2.0* encoder as the DAR backbone. Adding a single synthetic question per article yields the best overall performance, with substantial gains across all metrics compared to the model trained only on expert-annotated data (e.g., Recall@5 increases from 67.72 to 76.02). This confirms the usefulness of synthetic data in enhancing retrieval quality. However, as the number of synthetic questions increases beyond one, performance exhibits a decreasing trend. While additional synthetic queries still outperform the original dataset, the marginal gains diminish and in some cases slightly regress, suggesting that too many synthetic examples may introduce redundancy or noise. Overall, these findings highlight that carefully controlled amounts of synthetic data can significantly improve model performance, whereas excessive augmentation may be detrimental.

DAR (fine-tuning)	**Recall** (↑)		**nDCG** (↑)	**MRR** (↑)	**mAP** (↑)
	@5	@100	@100	@100	@100
Original dataset	67.72	82.92	69.00	67.53	62.57
1 synth. question	**76.02**	91.29	**77.59**	**78.01**	**70.15**
2 synth. questions	74.67	90.54	76.35	76.23	68.81
3 synth. questions	74.63	**91.37**	76.19	75.55	68.55
4 synth. questions	73.17	90.85	75.34	74.57	67.74

Table 12: Ablation on retrieval performance on the JuriFindIT validation set (DAR fine-tuning), by varying the number of synthetic questions per article added. Bold indicates the best score overall.

E.2 Batch size

We conducted an ablation study to evaluate the impact of batch size on retrieval performance under a contrastive learning setting (Table 13), using the best setting described in appendix E.1. In this scenario, each anchor–positive pair is contrasted against the remaining samples in the batch, which serve as negatives; to ensure training stability, we explicitly removed duplicate instances of the anchor/positive among the negatives.

Large batches. Since in-batch negative sampling benefits from having more negatives per query, larger batches can strengthen the training signal (Henderson et al., 2017). To make such batches feasible under GPU memory constraints, we employ GradCache (Gao et al., 2021), which recomputes activations during backpropagation and enables training with batch sizes that would otherwise not fit in memory.

Results show that performance varies with batch size, with the best overall results achieved at a batch size of 768. This configuration yields the highest scores across most rank-aware metrics (nDCG, MRR, and mAP), indicating that moderately large batches strike the best balance between providing sufficient hard negatives and avoiding excessive noise. Although a batch size of 512 attains the highest Recall@100, the 768 setting consistently outperforms others in terms of overall retrieval quality, and we therefore adopt it as our default training configuration.

DAR (FT)	**Recall** (↑)		**nDCG** (↑)	**MRR** (↑)	**mAP** (↑)
	@5	@100	@100	@100	@100
Batch size					
64	75.12	90.32	75.41	75.44	67.38
128	**76.73**	90.13	76.19	77.52	68.48
256 (GC)	74.61	89.93	76.65	77.30	69.46
512 (GC)	75.77	**91.84**	76.61	76.90	68.94
768 (GC)	76.02	91.29	**77.59**	**78.01**	**70.15**
1024 (GC)	75.81	91.20	76.40	75.81	68.83
2048 (GC)	74.74	90.28	75.42	74.77	67.78

Table 13: Ablation on retrieval performance on the JuriFindIT validation set (DAR fine-tuning), comparing different batch sizes. Training with batch sizes larger than 128 was only feasible through the use of GradCache (GC). Bold indicates the best score overall.

E.3 Pooling strategies

We compared two pooling strategies for obtaining sentence-level representations, namely using the contextualized `[CLS]` token and mean pooling over all token embeddings (Table 14). Results show that while the `[CLS]` token achieves slightly higher recall scores (76.39 vs. 76.02 at Recall@5 and 91.55 vs. 91.29 at Recall@100), mean pooling clearly outperforms it on rank-aware metrics. In particular, mean pooling yields the highest values for nDCG, MRR, and mAP, confirming its advantage when fine-grained ranking quality is required. Overall, the two methods perform similarly in terms of recall, but mean pooling provides more consistent improvements in ranking-sensitive evaluation.

DAR (FT)	**Recall** (↑)		**nDCG** (↑)	**MRR** (↑)	**mAP** (↑)
	@5	@100	@100	@100	@100
Pooling					
CLS	**76.39**	**91.55**	76.47	76.25	68.86
Mean	76.02	91.29	**77.59**	**78.01**	**70.15**

Table 14: Ablation on retrieval performance on the JuriFindIT validation set (DAR fine-tuning), by comparing CLS and mean pooling strategies. Bold indicates the best score overall.

E.4 GNN architectures

Table 15 reports the impact of different GNN architectures on retrieval performance when references between articles are either excluded (w/o refs) or included (w/ refs). Preliminary experiments showed that modeling the graph by merging reference edges into structural ones, i.e., $E_{struct} = E_{struct} \cup E_{ref}$, yielded better results. For this reason, all the reported comparisons have been carried out under this setting. Overall, GAT and GATv2 consistently achieve stronger results than the other architectures. In the setting without references, GATv2 attains the best Recall@5 and mAP, while GAT closely follows with competitive scores across all metrics. When references are included, GAT becomes the top-performing model, reaching the highest nDCG and MRR, while also maintaining strong recall. The comparison across scenarios highlights that incorporating references tends to slightly favor rank-aware metrics (nDCG and MRR), particularly for GAT, whereas in terms of recall the differences between the two settings remain small. These findings confirm that attention-based architectures (GAT, GATv2) are better suited than convolutional ones (GraphConv, SAGEConv) or heterogeneous transformers (HGT) for modeling the graph structure in our retrieval framework.

GIR	Recall (↑)		nDCG (↑)	MRR (↑)	mAP (↑)
	@5	@100	@100	@100	@100
w/o refs					
GraphConv	77.13	89.93	75.54	75.22	67.40
SAGE	77.74	89.17	75.99	75.88	68.81
HGT	75.29	**91.10**	75.82	76.02	67.93
GAT	77.17	91.04	77.94	78.10	70.87
GATv2	**77.80**	90.55	77.96	78.34	**71.00**
w/ refs					
GraphConv	75.86	90.62	75.37	75.13	67.19
SAGE	76.86	88.97	76.96	78.01	70.00
HGT	76.30	90.47	75.91	75.59	68.24
GAT	77.12	90.91	**78.00**	**78.67**	70.90
GATv2	76.90	90.87	77.80	77.67	70.77

Table 15: Ablation on retrieval performance on the JuriFindIT validation set (DAR fine-tuning), excluding and including references between articles (w/o refs, w/ refs sections), by varying GNN architectures. Bold indicates the best score overall.

F Evaluation metrics

Recall@k. Given a query q, let R_q be the set of relevant articles and F_q^k the top-k retrieved articles. Recall at cutoff k is:

$$\mathrm{Recall@}k(q) = \frac{|R_q \cap F_q^k|}{|R_q|}.$$

The final score is obtained by averaging over all queries:

$$\mathrm{Recall@}k = \frac{1}{|Q|} \sum_{q \in Q} \mathrm{Recall@}k(q).$$

This measures the proportion of relevant articles retrieved within the top-k results.

Mean Reciprocal Rank (MRR@k). Let rank_q be the position of the first relevant article for query q. Then:

$$\mathrm{MRR@}k = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\mathrm{rank}_q},$$

where $\mathrm{rank}_q \leq k$, otherwise the contribution is 0. This metric captures how early the first relevant article appears in the ranking.

Normalized Discounted Cumulative Gain (NDCG@k). For a ranked list of results up to position k, let $\mathrm{rel}_{q,i}$ be the graded relevance of the article at position i for query q. Then:

$$\mathrm{DCG@}k(q) = \sum_{i=1}^{k} \frac{2^{\mathrm{rel}_{q,i}} - 1}{\log_2(i+1)},$$

$$\mathrm{NDCG@}k(q) = \frac{\mathrm{DCG@}k(q)}{\mathrm{IDCG@}k(q)},$$

where $\mathrm{IDCG@}k(q)$ is the maximum possible DCG up to position k. The final score is averaged over all queries:

$$\mathrm{NDCG@}k = \frac{1}{|Q|} \sum_{q \in Q} \mathrm{NDCG@}k(q).$$

This metric measures the ranking quality by rewarding relevant articles that appear earlier, normalized against the ideal ordering.

Mean Average Precision (MAP@k). For query q, let $P_q(i)$ denote the precision at rank i, and define $\delta_{q,i} = 1$ if the article at rank i is relevant for q, and 0 otherwise. Then the Average Precision is:

$$\mathrm{AP@}k(q) = \frac{1}{|R_q|} \sum_{i=1}^{k} P_q(i) \cdot \delta_{q,i},$$

and the mean is:

$$\mathrm{MAP@}k = \frac{1}{|Q|} \sum_{q \in Q} \mathrm{AP@}k(q).$$

This combines precision and recall by averaging precision values at the ranks where relevant articles occur, providing a balanced measure of retrieval effectiveness.