Computer Law & Security Review 59 (2025) 106229

Contents lists available at ScienceDirect

Computer Law & Security Review: The International Journal of Technology Law and Practice

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

Enhancing legal document building with Retrieval-Augmented Generation ☆

Matteo Buffa [a,1], Alfio Ferrara [a], Sergio Picascia [a], Davide Riva [a,b,*], Silvana Castano [a]

[a] Università degli Studi di Milano, Department of Computer Science, Via Celoria, 18 - 20133 Milano, Italy
[b] Politecnico di Torino, Department of Control and Computer Engineering, Corso Duca degli Abruzzi, 24 - 10129 Torino, Italy

ARTICLE INFO

Keywords:
Legal document builder
Retrieval-Augmented Generation
Natural language processing
Legal information retrieval
Digital justice

ABSTRACT

Legal document building refers to the process of producing a legal textual document following a predefined schema with the support of digital, automated tools. Such systems must balance two fundamental requirements: providing targeted drafting assistance while preserving judicial autonomy and decision-making authority, and systematically leveraging existing legal document corpora to enhance consistency and quality in legal documentation. In this paper, we propose a document builder architecture, called JusBuild, designed to assist and support legal practitioners in drafting new legal documents. JusBuild supports the document assembly process by relying on a predefined legal document template and on a corpus of past legal documents. The key features of JusBuild are: (i) the use of a Conditional Random Field (CRF) model for the supervised segmentation of legal documents into functional sections according to a document template; (ii) a vector database storing segmented sections and their semantically meaningful vector representations for efficiently performing semantic search for suggestions retrieval; (iii) the suggestion, at drafting time, of relevant precedent sections retrieved from the vector database and of new, AI-generated sections, using a Large Language Model and Retrieval-Augmented Generation (RAG). A featuring design choice of JusBuild is the "human-in-the-loop" approach, which allows the user (judge) to exercise his/her decision-making freedom and full control in the formulation of the provision in working with the suggestions provided by JusBuild. Thanks to the flexible nature of the architecture, adaptable to a large number of legal contexts, with different document structures and legal matters, JusBuild makes contextualized content generation accurate and efficient for legal practitioners. The application of JusBuild to legal document building in the Italian legal context is discussed. JusBuild validation is provided by considering datasets that differ for document template, language, and judicial matter, to test its applicability and adaptability to different contexts.

1. Introduction

The legal field has traditionally been characterized by its reliance on vast amounts of documentation and manual processes carried out by legal practitioners in Courts, Parliaments, and other institutional bodies. Over the past years, significant effort has been put into the integration of digital technologies, especially those relating to Artificial Intelligence (AI), to improve accessibility of legal documents and efficiency in decision-making. Among the problems addressed, one main issue concerns the availability of appropriate tools to assist legal practitioners in the process of drafting new documents. Conventional methods of legal document building involve labor-intensive research, extensive review of legal precedents, and meticulous assembly of information into coherent documents. This process is time-consuming and can have significant consequences on the final outcome. These challenges could be overcome by implementing AI-driven solutions based on Retrieval-Augmented Generation (RAG), a recent and promising approach based on information retrieval and generative modeling. Leveraging various Natural Language Processing (NLP) techniques, a RAG system can retrieve relevant information from large corpora of legal texts and generate contextually appropriate content, thereby streamlining the document creation process.

In this paper, we present JusBuild, a comprehensive architecture for legal document building, which relies on several groundbreaking NLP techniques, in order to provide valuable support for legal practitioners

☆ This article is part of a Special issue entitled: 'JUSMOD23-SI' published in Computer Law & Security Review: The International Journal of Technology Law and Practice.

* Corresponding author at: Università degli Studi di Milano, Department of Computer Science, Via Celoria, 18 - 20133 Milano, Italy.
E-mail address: davide.riva1@unimi.it (D. Riva).

[1] Matteo Buffa is the sole author of Section 3 and collaborated with the other authors to Section 1, Section 5, and Section 7. Other sections are the product of the joint work of the other authors.

https://doi.org/10.1016/j.clsr.2025.106229

Available online 19 November 2025

in the act of drafting new legal documents. JusBuild spans from the segmentation of existing legal documents, to the generation and suggestion of relevant section contents for drafting the legal document currently under production. In particular, the *Document Segmentation Layer* of JusBuild deals with the segmentation of legal documents into functional segments, called *sections*, according to a document template. For this purpose, JusBuild implements a flexible text segmentation model based on Conditional Random Fields (CRF), to identify dependencies between clauses, which is easily adaptable to different document templates that individual jurisdictions may adopt. Textual content of document sections is then processed by an embedding model, capable of capturing the context of long documents; in this way, we obtain a semantically meaningful vector representation of sections, whose embeddings can be easily compared to perform semantic searches. The *Storage Layer* of JusBuild, based on a vector database, is responsible for collecting the text of the sections and the corresponding embedding vectors. For building a new legal document with JusBuild, the legal practitioner works on drafting a section at a time by receiving two sets of textual suggestions from JusBuild. A first set, called *Retrieved Sections*, is collected from past precedents, by exploiting semantic search over the vector database to retrieve those sections that are closer, most similar in terms of content, to the current section to be drafted. A second set, called *AI Generated Sections*, is provided by exploiting a Large Language Model (LLM) and RAG. A featuring design choice of JusBuild is the "human-in-the-loop" approach, which allows the user (e.g., the judge) to exercise his/her decision-making freedom and full control in the formulation of each document section, by selecting among the textual suggestions provided by the tool. To show adaptability to various legal contexts, JusBuild has been tested on two different datasets, which differ in the language of the legal documents, in the document template schema employed for segmentation, and in the judicial matter discussed. We show JusBuild at work on (i) the Italian legal context, by considering a dataset of first degree civil law judgments from 12 different Courts located in Northern Italy, and on (ii) the USA context, on an English-language dataset of court decisions from the Security and Exchange Commission. Evaluation of the components of the JusBuild architecture on both datasets was carried out separately, by first testing the *Document Segmentation* task and then the *Retrieval-Augmented Generation* task, respectively. The *Document Segmentation* has been evaluated only with respect to the ground truth defined in each dataset, which was annotated in both cases by teams of legal experts familiar with the matter discussed.[2] The *Retrieval-Augmented Generation* has been evaluated by the similarity of the retrieved and generated suggestions with the real active section in the dataset.

The paper is organized as follows. Section 2 is devoted to related work on legal document building and RAG. In Section 3, we discuss legal document building and jurisprudence practice to highlight requirements and challenges for document builder architecture design. In Section 4, we present the JusBuild architecture for RAG-enhanced legal document building. In Section 5, we discuss JusBuild application in the Italian legal domain. In Section 6, we present the validation process to test JusBuild functionalities on two different datasets of American and Italian court decisions. Finally, in Section 7, we draw conclusions and outline future research directions.

2. Related work

Our research focuses on the intersection between legal document building and RAG. This section reviews the main contributions and recent advancements concerning these two fields of study, which represent a significant evolution in the area of legal technology.

[2] For more information about the data, see the respective references [1] and [2].

2.1. Legal document building

Legal document building refers to the process of producing a legal textual document following a predefined schema [3] with the support of digital, automated tools. The task, even when entirely performed by humans, can be thought of as a sequence of three phases: (i) definition of the document format and structure; (ii) content draft of each part or section; (iii) editing of the document. Each of these phases is susceptible to automated support to a variable extent, ranging from interactive, human-in-the-loop approaches to fully automated services. For instance, document structure can be totally or partially based on predefined templates, possibly tailored to user needs. At the same time, editing can be supported by solutions ranging from error detection to automated rephrasing and text generation. Research distinguishes between document-oriented and issue-oriented approaches [4]. The document-oriented approach involves automatic selection and instantiation of document components based on user input. In contrast, the issue-oriented approach uses explicit legal rules, with their truth values determined by judicial justifications. Rules in issue-oriented approaches have been formalized in various ways, from decision trees [5] to interchange languages [6], with applications to Serbian case law [7] and to the GDPR [8]. Recent frameworks to legal document building include the presence of Knowledge Bases [9] and LLMs [10].

A fundamental component of document building is the task of Text Segmentation (TS), which involves the automated partitioning of a document into coherent segments, defined based on topical, semantic, structural, or functional criteria. The objective is to provide to the user only the sections of interest, relieving her from the burden of consulting the entire document. The approach to segmentation depends on the interpretation of coherence and can be categorized as either linear or hierarchical [11]. Linear methods view a document as a sequence of segments [12], while hierarchical approaches split segments at various levels, down to a specific granularity [13]. Additionally, TS methods can be region-oriented, focusing on detecting segment boundaries [14], or class-oriented, classifying each text unit (e.g., paragraph, sentence, clause, or word) into a segment type [1]. In the legal domain, both topical [15] and functional [16] approaches have been adopted, with the second being more suitable for isolating specific sections of interest, such as argumentative sections. In this regard, Conditional Random Fields (CRFs) have been proven to be successful at the task both on US Security and Exchange Commission judgments [1], and on Italian case law decisions [2].

It is crucial for a legal document building system to retrieve only the information pertinent to the document being built. For this reason, several legal information retrieval techniques have been proposed over the last years [17]. From boolean and rule-based approaches [18], to the exploitation of thesauri [19] and ontologies [20], researchers have recently focused on the use of Large Language Models [21], due to their capability of capturing the contextual representation of the text, an important aspect when dealing with domains having a peculiar language as in the case of the legal one. In particular, these techniques have been successfully employed for named entity recognition [22], and case law retrieval [23].

2.2. Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a paradigm enhancing text generation through the integration of relevant information retrieved from a domain knowledge source [24]. It compensates for the limited ability of LLMs in performing knowledge-intensive tasks, helping the model to deliver consistent and contextually relevant answers. A basic RAG pipeline consists of three main steps. First, the domain knowledge source is indexed, typically chunking the texts into semantically relevant segments and computing their vector representation through an embedding model. Then, at query time, text segments relevant to the user input are gathered, by employing retrieval techniques

ranging from simple keyword search to semantic similarity of vector embeddings. Finally, text generation is *augmented* expanding the user input with newly retrieved domain information.

Several RAG approaches have been successfully applied in a wide range of fields [25], including the legal one, where RAG has been applied to question answering and document building. In CBR-RAG [26] different embedding models and retrieval techniques are compared in order to perform legal question answering on Australian legislative and judicial documents. CaseGPT [27] is a framework for case-based reasoning which has been tested in both legal and medical domains. Other researches have addressed one of the main challenges encountered with legal texts, namely the complexity of the language used, affecting as a consequence also its generation. For example, in [28], the authors developed Chatlaw, a framework based on a combination of experts model and a multi-agent system, realized for generating professional case law consultations, while in [29] the objective of the research lies in returning answers that are informative and accessible to non-experts.

RAG is also being exploited as a support for the generation of new legal documents. For instance, LexDrafter [30] generates *Definition* articles for EUR-Lex legislative documents, based on pre-existing term definitions, while the authors of CLERC [31] provide a benchmark dataset for retrieving corresponding citations for a given piece of legal analysis and for generating the text of these citations.

2.3. JusBuild contribution

JusBuild aligns with the document-oriented approach to legal document building and employs advanced and established NLP techniques for text segmentation, information retrieval, and context-aware text generation. However, JusBuild diverges from existing systems in three critical ways. First, unlike more general legal QA systems or tools focused on generating ancillary document parts, like definitions or citations, JusBuild is tailored to the specific task of drafting operative sections of a judicial ruling. Second, it embodies a human-in-the-loop design philosophy that prioritizes the preservation of judicial autonomy in the document building process. Finally, the proposed architecture is a pragmatic solution deeply integrated into the procedural context of recent legal reforms in Italy and ongoing worldwide transformation, assisting the evolution of modern legal field with technological support.

3. Legal documents and legal document building

In order to address techniques for legal document building, we ground the process of drafting legal documents from a law and jurisprudence practice perspective, as well as on philosophical and ethical considerations, since this has an impact on the design requirements of a document builder architecture and its implications on the role of the user, with particular attention to the case of a judge user.

3.1. Legal documents, law and jurisprudence practice

Legal practitioners engage with legal systems primarily through document drafting, a complex activity that requires reasoned justification of decisions. This analysis focuses on judicial documents, i.e. measures adopted by judicial authorities that demand fully motivated legal reasoning. The *ratio decidendi* (reason for the decision) necessarily reflects both factual circumstances and legal interpretation, involving the reconstruction of events and their legal qualification in light of parties' claims. This decision-making process inherently involves discretionary balance in both parts. Indeed, the scope of judicial discretion has been extensively debated, particularly within the legal realism school of thought and critical legal studies, which emphasize the human element in judicial decision-making over purely deductive rule application. These approaches highlight judges' orientation toward equity and justice, recognizing law as interpretative activity rather than mechanical application [32–34]. The concept of "living law" [35] was formulated to encompass interpretations and applications of normative data across jurisprudence, doctrine, and administrative practices, extending beyond formal legal propositions to include social regulations observable in exchanges, customs, and group behaviors, even those not formally recognized by law. Legal realists notably argued that judges form preliminary decisions based on social interests, justice concepts, and personal factors, subsequently seeking legal justification in norms and precedents [36]. This creates a fundamental distinction between formal "paper rules" and operational "real rules" in judicial practice.

3.2. Realism perspectives on (building) justice through legal documents and activities

Jerome Frank [37], a prominent legal realist, emphasized the inherently subjective nature of judicial fact-finding. He argued that judges function as witnesses when evaluating testimony, inevitably filtering information through their personal perspectives, experiences, and unconscious biases. This subjectivity makes complete objectivity impossible, as judges cannot escape conditioning from their worldview. For Frank and other realists, legal certainty represents an unattainable myth: a "whole edifice of legal fictions" that humans construct to satisfy their need for security and stability. Legal uncertainty must be accepted as a permanent characteristic of human decision-making, where facts often become "conjectures" influenced by judicial opinion rather than objective reality [36]. This recognition transforms judging from a purely rational exercise into an art form that requires acknowledgment of its subjective elements. Reform demands training legal practitioners to understand their unconscious mechanisms and the substantial responsibility accompanying discretionary power. The judicial process becomes a competitive struggle between multiple possible representations of reality. Given these limitations of human judgment, AI offers potential value not as an end in itself, but as a tool for enhancing legal decision-making. AI systems could provide systematization and consolidation capabilities that offer guidance to decision-makers, potentially restoring some measure of confidence in legal certainty while acknowledging the inherent subjectivity of human judgment.

Recognizing the irreducible human dimension of judicial decision-making – shaped by personal, contextual, and interpretative factor – does not preclude the integration of technological tools within legal practice. On the contrary, it reinforces the need for systems that respect and support this complexity rather than reduce it. In this sense, artificial intelligence should not be viewed as a substitute for judicial reasoning, but rather as a complement capable of enhancing it. The challenge lies in designing architectures that preserve the discretionary and deliberative nature of judging while offering meaningful assistance in navigating and articulating legal arguments. From this perspective, the development of legal document builder systems must be guided by a dual imperative: to maintain the centrality of the judge as an autonomous agent, and to make use of the growing repositories of legal knowledge now available in digital form. The following section outlines the foundational requirements for such architectures, aiming to balance the interpretive richness of legal reasoning with the operational efficiencies enabled by (artificial, but still human-based) intelligent systems.

3.3. Requirements for a legal document builder architecture

Building on the recognition of inherent judicial subjectivity and AI's potential supportive role, document builder architectures for legal domains must preserve judges' central decision-making authority while providing effective drafting assistance. This approach acknowledges that fully automated systems are neither viable nor desirable given the essential human element in legal reasoning. Instead, interactive environments should support rather than replace judicial discretion. On the other hand, these architectures must effectively leverage the

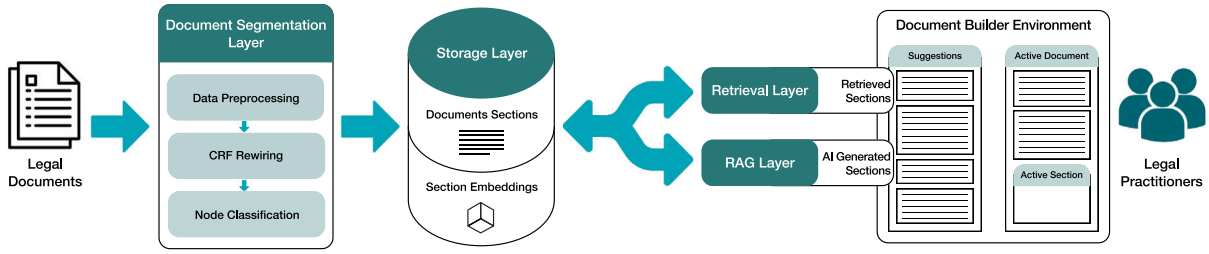

Fig. 1. The JusBuild architecture for legal document building.

vast corpus of legal documents continuously generated by courts, parliaments, and institutional bodies. By extracting relevant knowledge chunks from existing jurisprudence, AI systems can propose appropriate precedents and formulations for specific drafting tasks, such as judgment motivations. Digital transformation initiatives across jurisdictions have facilitated this capability through comprehensive legal information systems that ingest, process, and store documents in accessible digital formats. [38] AI-assisted document builders must, therefore, satisfy two fundamental requirements. First, they must provide targeted support for individual judicial decisions without compromising judicial autonomy. Second, they must systematically improve legal documentation standards by ensuring enhanced readability, consistency, and quality across judicial output. Additionally, these systems require feedback mechanisms that enable newly produced documents to contribute to an evolving knowledge base, creating iterative improvements in legal reasoning quality and accessibility.

4. The proposed JusBuild architecture

Based on the requirements discussed previously, we propose a document builder architecture, called JusBuild, designed to assist and support (without replacing) the judge in writing court orders and decisions. Right from the design stage, we followed a "human-in-the-loop" perspective, which allows the user (judge) to exercise his/her decision-making freedom and full control in the formulation of the provision, by selecting among the textual contents suggested by the tool, or by finding some elements of fact and law that contradict and change the consequences of the textual suggestions provided by JusBuild. The JusBuild architecture is depicted in Fig. 1. The architecture provides AI-based techniques to support the document drafting process by relying on a predefined legal document template and on a corpus of legal documents, properly segmented into sections according to the template, embedded, classified, and stored to support section-oriented suggestions retrieval and generation.

The *Document Builder Environment* is characterized by an *editing area* composed of a list of sections to be drafted, and a *suggestion area* where the user can retrieve pertinent suggestions to be re-used and optionally changed in the currently drafted section. The user is mainly focused on the editing area with the aim of writing and creating a new legal document according to a given document template. In the following, we refer to the new document under editing as the *active document*, and to the current section under drafting as the *active section*.

JusBuild is grounded on our previous work in the framework of the Next Generation UPP (NGUPP) project, funded by the Italian Ministry of Justice, aiming at providing AI and advanced information management techniques for the digital transformation of Italian legal processes and for digital justice in general. In particular, the JusBuild architecture proposed in this paper focuses on the *search-by-content* functionality of the initial version of document builder developed in NGUPP (see Castano et al. [10]) and provides the following extensions:

- a *Document Segmentation Layer* (based on the model proposed in [2]), for automated partitioning of legal document text into functional sections based on the considered document template, by using supervised classification techniques able to deal with data scarcity and label imbalance that characterize the legal text segmentation problem;
- a *RAG Layer*, for expanding the set of textual suggestions, that is, the set of *Retrieved Sections*, extracted from the legal document corpus and proposed to the user for editing the active section with a second set of *AI Generated Sections* pertinent for the active section. Integrating suggestions from relevant precedents with AI-generated suggestions gives the user a wider spectrum of ideas that might better contribute to the formulation of the active section. Employing a RAG-enhanced LLM generation ensures that the AI-generated content for a given active section remains relevant to the case at hand in the active document. This is achieved by leveraging the most similar previously retrieved document sections to construct a prompt that also includes the already completed sections of the active document, thereby minimizing deviation in the newly generated text.

4.1. Adopted notation

Before delving into the details of the architecture, we provide an overview of the adopted notation.

The whole JusBuild pipeline operates over a corpus C of past legal documents (e.g., court decisions), where each document conforms to a given template. A document template is articulated in a list of sections $x_1, \ldots, x_M$. We denote a document by d, an individual section by x_k and a sentence by s. Embedding vectors, used to represent these elements – documents, sections, and sentences – are denoted in boldface ($\mathbf{d}$, $\mathbf{x}_k$, $\mathbf{s}$). The JusBuild pipeline begins with the Document Segmentation Layer, which processes the corpus C by applying sentence-level embeddings $\mathbf{s} \in \mathbb{R}^D$ to split documents into semantically coherent sections. Then, each resulting section is embedded into a vector $\mathbf{x}_k \in \mathbb{R}^E$ using, also in this case, an embedding model. These Document Sections and their respective embeddings are stored in the *Storage Layer*, backed by a vector database to efficiently interact with the *Document Builder Environment*. At query time, the *Storage Layer* interacts with both the *Document Retrieval Layer*, which outputs the *Retrieved Sections*, and with the *RAG Layer*, which produces the *AI Generated Sections*.

In the *Document Builder Environment*, we define the legal document currently being drafted as d^*. This document is composed of M sections, of which the first $M-1$ sections, denoted as $x_1, \ldots, x_{M-1}$ are the already written sections, while the Mth section, $x^* = x_M$, is the active section, namely the section the legal practitioner is currently working on and for which suggestions are provided. Furthermore, each section x_k is associated with a label l_k drawn from a finite label set which categorizes its content.[3]

For the active section, JusBuild provides to the user two types of suggestions:

[3] Section labels are associated with the selected document template. For example, a basic template for judicial documents identifies three main sections, labeled respectively as `Introduction`, `Argumentation`, and `Conclusions`.

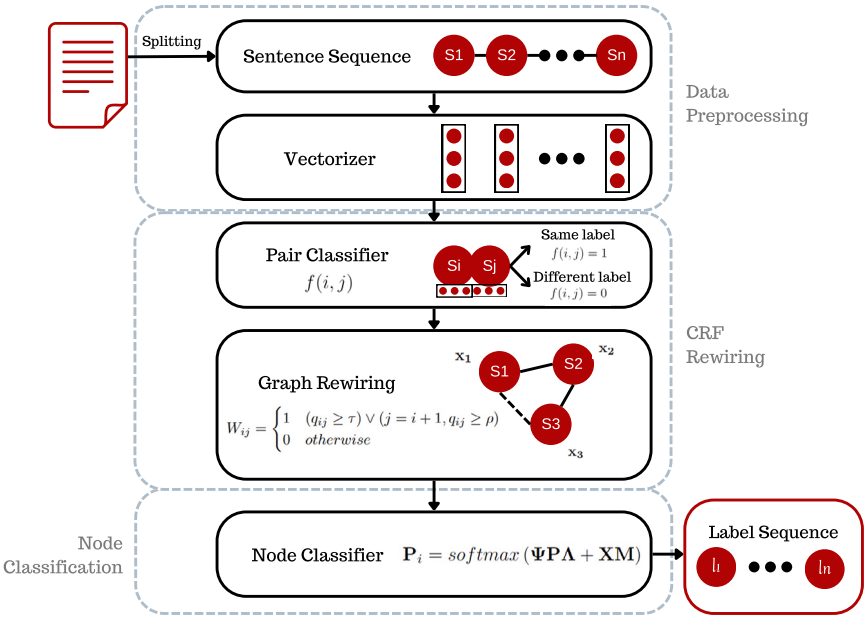

Fig. 2. Model structure for few-shot functional Text Segmentation.

- *Retrieved Sections* $\mathcal{RS}$: a set of suggestions $\hat{x}_M^1, \ldots, \hat{x}_M^T$, drawn from past legal documents in C. These are selected based on two criteria: (i) their labels match l^*, the label of the active section x^*; (ii) their preceding sections are semantically similar to and share the same labels as the written sections $x_1, \ldots, x_{M-1}$ of d^*.
- *AI Generated Sections* $\mathcal{GS}$: a second set of suggestions $\tilde{x}_M^1, \ldots, \tilde{x}_M^J$, produced by a generative LLM with RAG. These are created using a prompt that incorporates previously retrieved sections in $\mathcal{RS}$ to generate coherent and contextually relevant content.

4.2. Document Segmentation Layer

The *Document Segmentation Layer* aims at partitioning a textual document into coherent portions, called *sections* or *segments*. Although coherence may be interpreted from a variety of perspectives, for instance topical or structural, we argue that *functional* coherence, and thus functional segmentation, may be more relevant in the judicial domain, where parts of a document play clearly different functions and provide very different information. For instance, legal practitioners may benefit from systematically distinguishing `Introduction`, `Argumentation` and `Conclusions` in judicial documents. This step is made necessary by the fact that many documents, especially old ones which could not rely on a predefined standard template, may not come with a section structure.

We treat functional Text Segmentation as a supervised machine learning task. In this setting, each sentence is labeled according to its role in the document. This is based on the assumption that a sentence typically serves a single function in the economy of a document. However, this setup introduces two common issues: (i) data scarcity, due to the high time cost and necessary expertise for manual annotation, and (ii) label imbalance, as some sections are naturally shorter or longer than others. Further issues are represented by the specificity of legal jargon, which makes generic NLP approaches unfeasible, and by the possible discontinuity of sections expected from the articulated structure of judicial documents.

In order to counteract these issues, in Ferrara et al. [2] we proposed a segmentation model, depicted in Fig. 2, based on a Conditional Random Field (CRF) [39], a probabilistic model often used for segmenting and labeling sequential data. However, instead of using a standard linear-chain CRF, where each sentence is only connected to its immediate neighbors, we modify the CRF's structure using a Pair Classifier. The Pair Classifier takes a pair of sentences as input, embeds them into vector representations, and estimates the likelihood that they belong to the same section. Using this probability, denoted as q_{ij} for sentence pair (s_i, s_j), we adapt the CRF graph as follows:

- we have a connection between two sentences (s_i, s_j) if:
  - $q_{ij} \geq \tau$ (a high threshold) when the sentences are non-consecutive; or
  - $q_{ij} \geq \rho$ when the sentences are consecutive, i.e. $j = i + 1$;
- we do not have a connection between two sentences (s_i, s_j) in all the other cases.

The resulting CRF uses this revised graph structure to compute the probability that each sentence belongs to a specific section. This is captured by the following equation:

$$\mathbf{P}_i = softmax(\mathbf{\Psi P \Lambda} + \mathbf{SM}) \tag{1}$$

where $\mathbf{P}_i \in \Sigma^K$ (Σ^K being the simplex of dimension K) is the vector of probabilities of section labels for sentence i; $\mathbf{\Psi} \in [0;1]^{N \times N}$ denotes the updated transition matrix based on the rewired graph; $\mathbf{S} \in \mathbb{R}^{N \times D}$ is the feature matrix of all sentence nodes N in the graph, i.e. the matrix of sentence embeddings, and $\mathbf{\Lambda} \in \mathbb{R}^{K \times K}$ and $\mathbf{M} \in \mathbb{R}^{D \times K}$ are trainable weight matrices.

This approach addresses the challenges discussed before. Data scarcity is alleviated because each document of N sentences now yields $\binom{N}{2}$ sentence pairs, providing richer information for training the Pair Classifier. Label imbalance is tackled using two techniques: *loss weighting*, which consists in giving a higher weight during training to underrepresented sections; *positional knowledge injection*, which consists in assigning an initial probability equal to 1 to certain sections that always appear at specific portions of the document. For instance, if we hold the prior knowledge that documents must begin with a section called `Introduction` and end with a section called `Conclusions`, then we initialize $P_{1,Introduction} = 1$, with $P_{1,k} = 0$ for $k \neq Introduction$, and $P_{N,Conclusions} = 1$, with $P_{N,k} = 0$ for $k \neq Conclusions$. Discontinuity of sections is accommodated through flexible graph rewiring, allowing for connection of dispersed sentences. Finally, legal-specific language is handled using embedding models specifically trained for the legal domain.

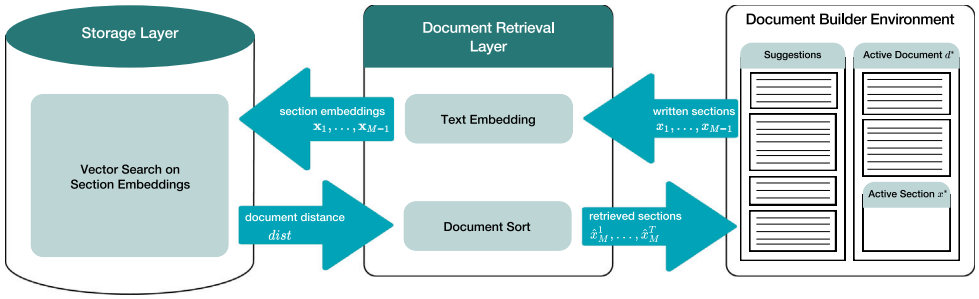

Fig. 3. Model structure for the Document Retrieval Layer.

Once sentences have been labeled with the corresponding section label, sections are reconstructed by concatenating their sentences in the order in which they appear in the document. Our model thus enables effective few-shot Text Segmentation in a complex scenario relying solely on semantically meaningful sentence embeddings or, alternatively or complementarily, carefully engineered sentence feature vectors.

4.3. Storage Layer and Document Retrieval Layer

The *Storage Layer* is meant to store document sections and the corresponding section embeddings, and implement CRUD operations on them. In particular, for each document section x_k, we store: the ID of the document d to which it belongs, the sequential position numbers of sentences of x_k in its document of origin, its textual content, its section label l_k and its embedding vector $\mathbf{x}_k \in \mathbb{R}^E$. To make the *Document Retrieval Layer* and the *RAG Layer* work at the section level, a necessary step is to embed entire sections in a vector space: instead of relying on vectors computed for the *Document Segmentation Layer*, which are computed at a higher granularity level and whose combination would be totally arbitrary, we adopt a second embedding model that can handle long context. Transformer models that have this feature are equipped with a particular kind of attention mechanism known as LSG (Local, Sparse, Global) attention [40]. By exploiting such models, we can capture the meaning of fragments of text as long as tens of thousands of tokens without losing part of the context and represent it in a vector space whose dimension E is in the order of 10^3. In our coarse example, for each (complete) document, we would have three vectors: $\mathbf{x}_{Introduction}$, $\mathbf{x}_{Argumentation}$, and $\mathbf{x}_{Conclusions}$. In the end, to enable effective and efficient query on such data, we rely on a vector database, namely, a database that internally implements vector indexing. Thanks to this feature, it is possible to perform fast vector similarity search over the database, that is, search of vectors that are closest to a query vector $\mathbf{v}$ according to a predefined distance metric δ. As there is a variety of applicable vector indices, the choice must carefully balance the quality of results, that is, the probability that the retrieved vectors are truly the closest (or most similar) to $\mathbf{v}$, as well as the search time. In JusBuild architecture, vector similarity search is at the basis of both *Document Retrieval Layer* and *RAG Layer*, as it is employed to retrieve sections of past documents that are similar to the sections of the active document.

Specifically for the *Document Retrieval Layer*, the aim is to retrieve those documents $d^i, i = 1, \ldots, T$, that contain sections $x_1^i, \ldots, x_{M-1}^i$ similar to written sections $x_1, \ldots, x_{M-1}$, where each x_k corresponds to a known section label l_k. The underlying idea is that the documents in the *Storage Layer* that are closer, in terms of content, to the active document d^*, could potentially present sections that are more relevant for drafting the active section x^*. Since each section search is independent from the others, we can execute the $M-1$ queries in parallel, accounting also for the fact that the order of sections may differ among documents. The interaction between the *Document Retrieval Layer*, the *Storage Layer* and the *Document Builder Environment* is depicted in Fig. 3.

The written sections $x_1, \ldots, x_{M-1}$ of d^* are first converted into vector embeddings $\mathbf{x}_1, \ldots, \mathbf{x}_{M-1} \in \mathbb{R}^E$, using the same embedding method described for the *Storage Layer*. These embeddings are then used as query vectors for the similarity search over the vector database. For each section labeled l_k, we retrieve the T closest matches from the *Storage Layer*. The similarity is measured using the inverse of a distance function $\delta(\cdot,\cdot) > 0$ so the best match for section labeled l_m is:

$$x_{l_m}^1 = argmax_{x_k^i : l_k = l_m} \delta^{-1}(\mathbf{x}_m, \mathbf{x}_k^i) \tag{2}$$

$$= argmin_{x_k^i : l_k = l_m} \delta(\mathbf{x}_m, \mathbf{x}_k^i) \tag{3}$$

where $x_k^i \in d^i \in C$.

This process is repeated for each of the $M-1$ sections to collect candidate segments from stored documents.

To find the documents in the *Storage Layer* that are overall the closest to d^*, we aggregate section-level similarities into a document-level score. For each match $x_{l_m}^i$, we store its metadata, namely, the document ID and the distance from the query vector $\mathbf{x}_m$. We then group all section matches by document ID and compute a total distance by summing the distances of the matched sections. If a document d^i does not contain a required section x_m^i with label l_m, we penalize this absence by assigning it the maximum distance observed for that section, denoted δ_{Tm}. Formally, the total distance for document d^i is computed as:

$$dist(d^i, \{x_1, \ldots, x_{M-1}\}) = \sum_{m=1}^{M-1} \hat{\delta}_{im} \quad \text{with} \quad \hat{\delta}_{im} = \begin{cases} \delta_{im} & \text{if } x_m^i \in d^i \\ \delta_{Tm} & \text{otherwise} \end{cases} \tag{4}$$

Notice that this is a lower bound on the sum of the real distances between the section embeddings of the active document and document d^i. While this approach may not compute the exact minimum distance (or, equivalently, the exact maximum similarity), it offers a highly efficient alternative by leveraging the power of the vector database, thus enhancing the potential of the whole architecture.

Finally, the T documents with the lowest total distances are selected. From each document, we extract the target section labeled with l_M (e.g., `Conclusions`), which composes the *Retrieved Section* set $\mathcal{RS} = \{\hat{x}_M^1, \ldots, \hat{x}_M^T\}$, passed to the *Document Builder Environment* and the RAG Layer, respectively.

4.4. RAG Layer

The last component of the JusBuild architecture is the *RAG Layer*, which is in charge of expanding the set $\mathcal{RS}$ of retrieved sections provided to the user with new, AI-generated sections. Its functioning exploits a generative LLM, used to produce new suggested sections for the active section x_M, namely $\tilde{x}_M^1, \ldots, \tilde{x}_M^J$. In order to obtain results that are at the same time coherent with the active document d^*, i.e., with its sections already drafted and completed $x_1, \ldots, x_{M-1}$, and reflective of the relationship between sections in similar cases, we design a prompting mechanism built upon the results of the *Document Retrieval Layer*. In particular, among the retrieved documents, we pick the document d^1, namely the document most similar to the active

document, and we provide it as "best example document" to the LLM. This technique is called *one-shot learning*, and consists of providing to a machine learning model one example of the task it is supposed to achieve. The reason for selecting the best-matching retrieved document is two-fold: on one hand, it is preferable over giving no examples since it provides the model with a more defined idea of what kind of output it is expecting to produce; on the other hand, providing multiple document examples would burden the LLM with an excessively long prompt, thus worsening computing performance and, potentially, output quality. Formally, given the retrieved sections $\hat{x}^1_M, \dots, \hat{x}^T_M$ and the corresponding documents $d^1, \dots, d^T$, we select the most similar pair, i.e. $(d^1, \hat{x}^1_M)$, to be used as "best example document" and its ground truth. The LLM is given a prompt with the following analogical form (modulo translation in the appropriate language):

> *Given the following document:* $\langle d^1 \rangle$. *Given its next section:* $\langle \hat{x}^1_M \rangle$. *Generate the next section for the following new document:* $\langle x_1, \dots, x_{M-1} \rangle$.

Repeating the request J times, possibly with model parameters that enhance the variability of the output, yields, as a result, a set of *AI Generated Sections*, $\mathcal{GS} = \{\tilde{x}^1_M, \dots, \tilde{x}^J_M\}$, which are provided as suggestions to the user inside the *Document Builder Environment* to be freely employed in the active section drafting process. The set of *AI Generated Sections* is displayed separated from the set of *Retrieved Sections* in order for the user to clearly understand the provenance of each kind of suggested sections and better decide if and how to (re)use them.

4.5. Document Builder Environment

The *Document Builder Environment* is the interface through which the user (judge) interacts with JusBuild architecture. The user is presented with a text editing area, divided into the sections of the adopted document template. For instance, it may be divided into `Introduction`, `Background`, `Claims of the parties`, `Argumentation` and `Decision` sections, as in the document template discussed in [41]. Organizing legal document according to a given template is motivated in the literature (for further references, see [42]) by the need to provide structure to legal documents that do not have a standard one, such as court decisions, as well as by the need to provide a guidance in the process of building legal documents, with the aim of improving readability and usability without harming the freedom of the judge drafting the document. Besides the adoption of a document template, the editing area of JusBuild allows the user to draft all sections by himself/herself, as in a traditional text editor. The distinguishing feature of JusBuild resides in the suggestion area of the *Document Builder Environment* that provides pertinent suggestions for the active section. Here, both *Retrieved Sections* and *AI Generated Sections* are shown to the user, who can thus choose whether to import the suggestions as they are, or import and edit them, or continue drafting the document without using suggestions. We deem this "human-in-the-loop" approach crucial to avoid unduly restricting the judicial power of the judge.

5. JusBuild at work and applicability issues

In this section, we show JusBuild at work considering a case study of legal document building developed within the framework of the *NGUPP* Italian project. Then, we provide considerations about applicability of the proposed document builder functionalities, also in light of the recent EU AI-Act regulation.

5.1. JusBuild at work in Italian legal context

The NGUPP project's document builder architecture emerges from the evolving landscape of Italian civil procedure, where recent reforms mandate electronic filing and standardized judicial document templates, creating a shared data ecosystem where each judicial act contributes to collective jurisprudential knowledge rather than existing as an isolated document. This digital transformation necessitates AI systems that preserve judicial autonomy by positioning judges as sole authors responsible for validating data and ensuring argumentative consistency. JusBuild architecture aligns with both practical legal requirements and emerging regulatory frameworks that emphasize human oversight in judicial AI applications, functioning as an intelligent assistant that enhances judicial capabilities while preserving the essential human element in legal reasoning. Although significant potential exists for expansion to include jurisprudence from further national and international judicial bodies such as the Italian Corte di Cassazione, Constitutional Court, European Court of Justice, and European Court of Human Rights, what we present here is a case study tailored to civil trials in first-degree courts ("Tribunale") but applicable also to second-degree courts ("Corte d'Appello") as well as on criminal law.

Another aspect to carefully consider in this realm is the relevant regulation on AI and data protection existing in the European Union. Under the EU AI Act's regulatory framework, the document builder falls within the high-risk category as an AI system assisting judicial authorities in legal research and interpretation, necessitating stringent compliance measures that acknowledge its central function in judicial decision-making rather than merely administrative support. The regulation mandates that final decision-making remains human-led, with judges maintaining ultimate responsibility for legal determinations, requiring impact assessments on fundamental rights before deployment, and robust human oversight arrangements that consider potential impacts across different legal competencies and case scenarios. JusBuild presents ethical implications requiring careful monitoring, though these are comparable to existing legal databases that provide automated citation and content-concept searching capabilities, with its primary value lying in enhancing precedent citation accuracy and supporting coherent conclusion generation while maintaining judicial accountability. Ongoing compliance verification requires collaboration between deployers, judges and Data Protection Authorities,[4] ensuring alignment with both technical requirements and fundamental legal system guarantees, with the system's success depending on balancing technological capability with preservation of judicial independence and human rights protection in the context of Italy's civil law tradition that does not formally follow *stare decisis* (the principle of following as closely as possible precedents established by previous judicial decisions when deciding similar cases) but increasingly values superior court precedents in contemporary legal practice.

To effectively support document building in this context, we first adopted a document template for the class of documents at hand, namely, the Italian civil law judgments. We recall that customization is an essential feature of our tool as different legal contexts involve different conceptions of a legal document. The document template for Italian court decisions, defined in collaboration with the legal partners of the NGUPP project, is inspired by best practices in judicial writing and tailored to the Italian context [42], and organizes a judgment into the following five sections (with corresponding labels):

- `Court and parties` (l_1), providing information about the court, the panel of judges and the parties in the trial;
- `Background` (l_2), relating to the proceedings of the trial and to the reconstruction of the facts;
- `Argumentation` (l_3), containing claims made by the plaintiffs and counterclaims made by the defendants;

[4] As mandated by GDPR, article 57.

- `Motivation` (l_4), providing reason(s) underlying each decision about an individual claim;
- `Decision` (l_5), providing the final decision(s).

The *Document Segmentation Layer* is trained to recognize the target section labels $l_1, \ldots, l_5$, and the document template in the *Document Builder Environment* is organized according the above five sections.

The first three sections (`Court and parties`, `Background`, and `Argumentation` from the parties) are largely descriptive and can be automatically or semi-automatically populated using information from the official electronic case files, such as transcripts from hearings and depositions. Thus, the primary challenge for a judge lies in drafting the `Motivation` and `Decision` sections. These are the most labor-intensive and knowledge-intensive parts of the judgment. Acknowledging the distinct nature of these two sections, we made a strategic design choice. The `Motivation` section represents the core of judicial reasoning and is unique to each case. To preserve the judge's full intellectual autonomy in this critical task, we intentionally left the drafting of this section to the user, thus focusing our experiments on the `Decision` section. While the reasoning is unique, the operative language of a decision often follows established legal conventions and precedents, relating for instance to penalty applied and compensations for expenses in light of the motivations. By providing relevant examples and suggestions for this section, our tool can significantly reduce drafting time and improve consistency.

To show the drafting of a new legal document using JusBuild, we consider a case study developed in the context of the NGUPP Italian project, working on a dataset of 50 first-degree civil judgments from 12 courts in Northern Italy, all concerning the matter of unfair competition. After extracting the plain text from these documents, we segmented each one according to the five-section schema previously described. In the following section, we will show how JusBuild uses the *Document Retrieval Layer* and the *RAG Layer* to similarity-based retrieval and AI-generation of section suggestions for an active `Decision` section to help formulate the final ruling, by exploiting the dataset of 50 first-degree civil judgments. The quantitative evaluation of document segmentation, section retrieval, and RAG generation capabilities of JusBuild will be discussed in Section 6.

5.2. Drafting a Decision section with JusBuild

In this section, we describe the application of JusBuild for the drafting of a new `Decision` section in a new case law document regarding `counterfeiting of machinery`.

As stated in the previous subsection, the *Document Builder Environment* will provide the user with the first three sections already populated with the trial data extracted from the case files[5]:

> `Court and parties`: *COURT OF M. - CIVIL SECTION - FIRST INSTANCE Case No.: 2024CV1847, Date of Decision: March 15, 2024, K. S.p.A. (plaintiff, represented by Attorney A.B.) v. F. S.r.l. (defendant, represented by Attorney C.D.)*

> `Argumentation`: *By writ of summons served on 22 November 2004, K. S.p.A., sued F. S.r.l., requesting, by finding and declaring that F. S.r.l., in the person of its legal representative pro tempore F., F. and E., have carried out acts of unfair competition against K. S.p.A.: - F. S.r.l. to pay compensation for all consequential and inherent damages for the aforementioned acts of unfair competition, to be quantified and settled in a separate proceedings, pursuant to Article 278 of the Code of Civil Procedure; - order the publication of the judgment in a national newspaper and in a local newspaper, with the costs to be borne jointly and severally by the defendants. In particular, the plaintiff claimed that the defendants were liable for unfair competition under Article 2598(3) of the Civil Code for misappropriation of the plaintiff company's clientele and under Article 2598(2) of the Civil Code for unlawful appropriation of the merits of the plaintiff company's products. [...]*

> `Background`: *At K. S.p.A., F. and E. had been employees with the task of managing, with a large degree of autonomy, relations with customers and taking care, in every aspect, of the sales of the machines, their spare parts and other K. S.p.A. - they had thus become acquainted with the machines manufactured and sold by the plaintiff, their technical characteristics, their components and the consumables necessary for their operation, the organization of K. S.p.A.'s production organization, which is directly involved in the conception and design of the machinery, the drafting of drawings and technical specifications thereof, as well as the customers of the plaintiff company and the conditions, in particular economic conditions, governing the contractual relationship with it. [...]*

We remark our scenario, in which the judge manually compiles a draft of the *Motivation* section as the one exemplified by the following excerpt:

> `Motivation`: *[...] F. S.r.l. only carries out service activities for the repair of cleaning machinery, while K. S.p.A.'s main activity is the manufacture and sale of the machinery, which is not carried out by F. S.r.l., consequently, there can be no question of unfair competition since the defendant company's activity is that of mere marketing and not that of production, which is the exclusive responsibility of the plaintiff company. The present case, as has already been said, 'does not concern the legitimacy or otherwise of the production of "compatible" and "non-original" parts whose legitimacy of "production" in itself is guaranteed by the principle of free competition (see on this point Cass., sec. I, 28/10/1998, no. 10739) but only the possible forms of unfair competition'. As already mentioned, the plaintiff company complains that the defendants are liable for unfair competition pursuant to Article 2598(3) of the Civil Code in that they allegedly marketed spare parts identical to those of K. S.p.A. However, this complaint made by the plaintiff is unfounded since the recent orientation of the jurisprudence of legitimacy states that: 'The manufacturer of spare parts (rectius F. S.r.l.) is allowed to use the trade mark in order to indicate the intended purpose of the goods it offers' (Cass., sec. I, 21 June 2000, no. 8442) and again: "The spare parts manufacturer (rectius F. S.r.l.) has the option of using another person's trade mark together with its own in order to indicate the intended purpose of the goods it offers" (Cass., sec. I, 28 October 1998, no. 10739). [...]*

After completing the draft of the `Motivation` section, our active document in JusBuild has the first four sections filled in, and the `Decision` section becomes now the active section. For drafting this section, the judge has a choice: he/she might go on drafting the active section manually, or he/she can ask for suggestions to JusBuild. In this latter case, JusBuild retrieves as suggestions the most relevant `Decision` sections from the past judgments in the considered dataset that are overall similar (in terms of `Court and parties`, `Background`, `Argumentation` and `Motivation` sections) to the active document. We retain also the `Court and parties` similarity score since it may be relevant in some specific cases, everything else equal.[6] For the sake of simplicity, hereafter we show the top-2 retrieved `Decision` sections, ranked by their similarity score:

[5] The first three sections can be automatically composed by exploiting the civil information system functionalities for extracting tyrial data of interest from the official electronic case files, such as transcripts from hearings and depositions. We note that the case study has been conducted on legal documents written in Italian; retrieved textual suggestions reported in the following examples have been manually translated into English for the purpose of readability and understanding.

[6] For instance, special legislations limited to a sub-national geographic area.

$\hat{x}^1_M$: *The plaintiff's claims are unfounded and cannot be upheld, for the reasons in fact and law set out below The Court of ***, in the person of the Single Judge, Ms ***, giving final judgment, and rejecting all contrary requests, objections and preliminary inquiries, (1) Dismisses the claims brought by S.T.P. SPA against A.P. S.R.L. (already A.A. SRL) and R.L. (2) ORDERS S.T.P. SPA to pay A.P. S.R.L. (already A.A. SRL) and R.L. the costs of the proceedings, which it settles, for each of the defendants, as regards the ordinary proceedings, in the amount of Euro 9275.00 for lawyers' fees, plus general expenses at the rate of 15% of the fees, VAT and CPA as required by law and, as regards the interlocutory proceedings, including the complaint phase, Euro 2552.20 for administrative fees.*

$\hat{x}^2_M$: *The defendant's claim for revocation for non-use of the plaintiff's registered trade mark is unfounded. The application for revocation is therefore dismissed. The defendant's objection of lack of standing on the part of the plaintiff for the declaration of invalidity of the brand with reference to classes A and B is consequently declared partially invalid, as the graphics used by the plaintiff are irrelevant for the purposes of the infringement proceedings. The application for a declaration of infringement made by the plaintiff is therefore granted. [...]*

Under the hood, JusBuild also sends the first retrieved section $\hat{x}^1_M$, together with the original document d^1, to the *RAG Layer*. Using this information to compose a prompt submitted to LLM as described in Section 4.4, JusBuild generates a further set of suggested sections (which are labeled AI generated sections), two of which are reported hereafter:

$\tilde{x}^1_M$: *The Court of ***, in the case of unfair competition between K. S.p.A. and F. S.r.l., rejected the claims put forward by the plaintiff, ruling that the conduct of F. S.r.l. does not constitute unfair competition under Article 2598 of the Italian Civil Code in relation to the misappropriation of the merits of K. S.p.A. and the diversion of customers. In particular, it emerged that the activity of F. S.r.l. was limited to carrying out market research without appropriating confidential information. The Court also observed that, although F. S.r.l. had used identification numbers of K. S.p.A., such practices fell within the permissible legality within the principle of free competition. Therefore, F. S.r.l. did not breach any duty of confidentiality and did not engage in unfair competition. Lastly, the request for publication of the judgment in the newspapers was reserved for a possible separate judgment, while the costs of the litigation are borne by the losing party. The plaintiff's claims, therefore, were unfounded and the Court pronounced their rejection, confirming the correct conduct of F. S.r.l. and its representatives.*

$\tilde{x}^2_M$: *Based on the description provided, it appears that the issue concerns the liability of the company F. S.r.l. for unfair competition pursuant to Article 2598 n. (3) of the Italian Civil Code and Article 2598 n. (2) of the Italian Civil Code, since the company F. S.r.l., which had been established after the period of dependence of the partners F. and E. on the company K. S.p.A., allegedly used the knowledge acquired at that time concerning the machines produced and sold by the parent company, their technical characteristics, the organization of the production of K. S.p.A.'s production organization, as well as its customers and its contractual terms and conditions. The company F. S.r.l. contacted K. S.p.A.'s main English customer, H., which was, in practice, the exclusive distributor in England of the parent company's products, presenting itself as a multi-brand company and promoting its business. Finally, it is not clear what the outcome of F. S.r.l., since it is not specified whether the court has recognized liability for unfair competition or not and, therefore, whether the company must pay the costs of litigation or not.*

The JusBuild suggestion capabilities thus enable the judge to acquire domain-specific knowledge, prominent/essential for completing the active section. JusBuild facilitates knowledge extraction by identifying relevant textual excerpts and by establishing conceptual connections between the active section and existing jurisprudential materials. This retrieval process operates through the *Storage Layer* and the *Document Retrieval Layer* of the architecture, which maintain bidirectional linkages between suggested content fragments and their source documents, thus enabling the user to fully understand the information flow and to discriminate between relevant and less/not relevant information. This contextual awareness prevents decontextualized application of legal reasoning and maintains the integrity of jurisprudential precedent.

The expansion of retrieved sections through AI-generated content ($\tilde{x}^1_M$, $\tilde{x}^2_M$) reveals both the potential and limitations of current generative approaches in legal document building. While the generated suggestions for the active section effectively synthesize factual elements and provide coherent case overviews, they exhibit systematic gaps in jurisprudential grounding, specifically lacking references to established case law and doctrinal authorities that constitute essential components of legal reasoning in both civil and common law systems. This deficiency highlights a critical architectural consideration: generated content serves optimally as intermediate scaffolding for factual analysis rather than complete legal argumentation. The system's strength lies in synthesizing complex factual scenarios and identifying their legal relevance, particularly in the "in fact" dimension of judicial reasoning. However, the "in law" dimension requires integration with authoritative sources that current generative models cannot autonomously provide. We highlight the task of combining these two dimensions in a comprehensive RAG system as a fundamental direction for future research in this field.

These findings suggest that optimal performance can emerge from complementary deployment of retrieval and generation capabilities. Retrieved suggestions provide authoritative grounding and precedential context, while AI-generated ones offer synthetic clarity and factual organization. The judge is engaged with a supervisory, curatorial and deliberative role: selecting appropriate authorities, verifying factual syntheses with appropriate analogies and dissimilarities, ensuring doctrinal completeness through manual supplementation where AI-generated content exhibits gaps in jurisprudential reference, and finally integrating all selected components in a unified deliberation.

6. Quantitative evaluation of the JusBuild architecture

To validate the effectiveness and adaptability of the JusBuild architecture, we conducted a comprehensive evaluation. Our goal was to validate the performances of three layers of the architecture.

1. Can our *Document Segmentation Layer* effectively identify the functional sections of legal judgments across different languages and annotation schema?
2. How well does the *Document Retrieval Layer* collect relevant precedent sections to assist in drafting a new document, specifically the *Decision* section?
3. Does the *RAG Layer* produce text that is coherent, contextually relevant, and potentially useful to a judge who is drafting the document?

To assess segmentation accuracy (RQ1), we will measure the F_1 score of our model and compare it against a widely-used Conditional Random Field (CRF) baseline, to rigorously quantify our model's performance relative to a strong, established method in the field. Next, to evaluate retrieval quality (RQ2), we will calculate the Mean Reciprocal Rank (MRR) of the most relevant precedent. This metric is ideal for measuring the practical utility of a ranking system, as it directly reflects how quickly a user can find useful information. Finally, to determine the usefulness of the generated text, we will test different Large Language Models (LLMs), comparing generated text against the best-retrieved examples, and isolating the impact of our RAG approach.

Table 1
Annotation schema for US court decisions.

Label	Definition	# Sentences
Introduction	Court, judges, case citation, parties	1235
Background	Procedural history, relevant facts, parties claims	6573
Analysis	Court discussion and opinions	17 479
Concurrence or Dissent	Opinions of concurring or dissenting judges	454
Footnotes	Information about references	2261
Appendix	Supplementary materials	561
Conclusions	Court decision	1656

Table 2
The annotation schema for IT court decisions.

Label (in Italian)	Definition	# Sentences
Court and parties (Corte e parti)	Court, judicial panel, parties	750
Background (Antefatto)	Background information	1514
Argumentation (Domande)	Claim(s) and argumentation of the parties	260
Motivation (Motivazione)	Reason(s) for the final decision(s)	2559
Decision (Decisione)	Final decision(s)	458

6.1. Datasets

We used two distinct datasets to evaluate the architecture's adaptability. The first dataset (US) [1] consists of 173 court decisions involving trade secrets from the online Court Listener.[7] The documents have been manually annotated by the paper authors using a seven-label schema presented in Table 1. Of the seven, two labels were underrepresented in the dataset: for this reason, the original study filtered out documents presenting these uncommon sections during the evaluation. In contrast, we decided to retain the full dataset to test our model's robustness.

The second dataset (IT) [41] is composed of 38 case law decisions on the matter of unfair competition, retrieved from 12 courts in Northern Italy. A group of 9 legal experts have manually annotated the documents according to the annotation schema presented in Table 2. The schema is specific to the Italian civil trial procedure and includes five sections. Given the inherent complexity and subjectivity in segmenting legal texts, we adopted a perspectivist approach, treating all annotations from the experts as equally valid to handle potential disagreements.

6.2. Tasks and metrics

The use of these two distinct datasets is a deliberate experimental choice designed to test the architecture's robustness and adaptability. While the legal content, language, and annotation schemas differ significantly, both datasets share a common underlying data structure, which is key to our model's cross-corpus applicability. In both datasets, each document is broken down into a sequence of sentences. The evaluation of the *Document Segmentation Layer* involves classifying each individual sentence with a functional label (e.g., Motivation, Conclusions). A contiguous block of sentences assigned the same label constitutes a functional segment or section, formally defined in the data by its character start and end offsets. The data granularity shifts for the subsequent layers of the JusBuild architecture. While segmentation operates at the sentence level, the evaluations of the *Document Retrieval* and *RAG Layers* operate at a higher level of abstraction. For these tasks, we treat the entire aggregated text of each section as a single unit. Specifically, our evaluation focuses on retrieving and generating suggestions for the final "Conclusions" (US) and "Decisione" (IT) sections.

We evaluated each layer of the architecture with appropriate metrics.

- *Document Segmentation Layer*: we measured sentence-level F_1 score to assess the accuracy of our segmentation model. We compared our approach against a standard linear-chain Conditional Random Field model, a widely adopted approach for the Text Segmentation task.
- *Document Retrieval Layer*: to measure the performance on the retrieval task, we employ different metrics to compute the relevance of the retrieved sections, and the quality of the ranking. In particular:

    1. to evaluate the relevance we compared the content of the sections retrieved against the true section of a document in terms of ROUGE-L [43] and BLEURT [44]. ROUGE-L measures sentence-to-sentence similarity based on Longest Common Subsequence (LCS) statistics; its scores range between 0, in case there is nothing in common between two sentences, and 1, when two sentences are exactly the same. BLEURT, instead, is an evolution of the BLEU metric, exploiting the contextual word representations of BERT [45] and a novel pre-training scheme; its scores range between −1 and 1, and it is typically employed for the comparison of different systems on the same task. The values of the two scores at which the performances of a model can be considered satisfactory may vary a lot depending on the task being solved, with state-of-the-art models generally reaching a ROUGE-L score above 0.2 and a BLEURT score above −0.5 [46–48];
    2. to evaluate the quality of the ranking, we employ the Mean Reciprocal Rank (MRR). This metric tells us, on average, how high the most relevant document is ranked in the list of 10 retrieved sections. A higher MRR (closer to 1.0) indicates better retrieval performance.

- *RAG Layer*: we used again ROUGE-L and BLEURT to compare the generated text against the ground-truth sections. We conducted three comparisons: (i) the performance of two different LLMs (the proprietary GPT-4o-Mini vs. the open-source Mistral-7B), (ii) the quality of generated text versus the best-retrieved text, and (iii) the impact of RAG (generation with retrieved context) versus generation without it.

6.3. Experimental setting

All our experiments are executed on a Zorin 16.3 machine with 32 GB RAM and 16 CPU cores. The configuration of the architecture involves a set of features independent of the document metadata, in particular of the document language, while another set depends, in

[7] www.courtlistener.com

Table 3
Configuration features, divided into those that are in principle metadata-independent and those that depend on the dataset metadata. Note: nowadays, most widespread LLMs can handle multiple languages, and therefore might be considered as a metadata-independent feature, but we cannot consider this, in principle, as a universal guarantee.

Layer	Metadata-Independent features	Metadata-Dependent features
Document Segmentation	Pair Classifier, τ and ρ thresholds	Sentence splitter, vectorizer
Storage and Section Embeddings	DBMS, distance metric δ, vector index	Embedding model
Document Retrieval	Number T of results	Section l_M of interest
RAG	Number J of results	LLM, prompt language

principle, on the metadata. Metadata-dependent features, for instance, the transformer models involved for embedding and the prompt employed for generation, are strictly dependent on the corpus language and the section currently being drafted; other features, such as the hyperparameters of the Document Segmentation Layer, as well as the number of *Retrieved* and *AI Generated Sections*, are independent of the corpus considered. The two classes of configuration features are summarized in Table 3.

In the following, we provide, layer by layer, the feature configuration employed for our experiments:

- in the Document Segmentation Layer, we adopt a feedforward neural network with a single hidden layer of dimension 256 and ReLU activation function as Pair Classifier, trained with AdamW algorithm and cyclic learning rate in the interval $[10^{-3}; 10^{-2}]$. We experiment with threshold $\tau \in \{0.65, 0.80, 0.95\}$ and $\rho = \frac{\tau}{2}$, which mimic the cases of a dense, intermediate and sparse graph respectively. Sentences are split whenever a newline or a dot occurs (excluding common abbreviations), and we employ, as sentence vectorizers, Legal-BERT[8] for English documents, and Italian-Legal-BERT[9] for IT documents;
- in the Storage Layer, we adopt Chroma[10] as DBMS, L^2 as distance metric, and Hierarchical Navigable Small Worlds (HNSW) [49] as index. For Section Embeddings, we employ LSG-BART[11] as embedding model, given its effectiveness for both US and IT documents;
- in the Document Retrieval Layer, the number of retrieved sections is set as $T = 10$, while the section of interest is referred to as "Conclusions" in the English documents and "Dec" (for "Decision") in the IT documents;
- finally, in the RAG Layer, we set the number of *AI Generated Sections* as $J = 1$, experimenting with two LLMs, a proprietary model, GPT-4o-Mini,[12] and an open-source model, Mistral-7B [50].

6.4. Results

In the following section, we report the results for each layer independently.

Document Segmentation Layer. In Table 4, we report the sentence-level F_1 scores for the text segmentation task: while the different models were identically run on the IT data, in the case of US data the baseline performance reported in [1] takes into account only 5 sections, excluding "Concurrence or Dissent" and "Appendix" due to data scarcity. Thus, we evaluate our model in two scenarios for US data: one in which we restrict the dataset to the 5 labels used in the original work, and another in which we consider all the 7 sections.

It can be noticed that our model outperforms the baseline on IT data, while underperforming on US data. This difference could be attributed to the higher class imbalance of the US dataset, for instance with the section "Analysis" appearing much more frequently than the others, but also to the chosen sentence vectorizer which, though tailored to the legal domain and better than general-purpose ones, may still be unable to capture information essential for the task, which may be better extracted via careful feature engineering. Nevertheless, our approach delivers consistent performances irrespectively of the amount of labels involved in the task: F_1 scores on US data are similar for both scenarios, despite the scarcity of the two additional labels being addressed as an issue in the original work.

Table 4
F_1 scores for the Document Segmentation Layer of a linear-chain CRF and of our approach at different parameter configurations, computed on the IT and US court decision datasets. The best result on each dataset is highlighted in bold, the second best in italic.

Model	IT	US (5 sections)	US (7 sections)
Baseline (linear-chain CRF)	0.622	**0.717**	–
Ours ($\tau = 0.65$)	0.620	0.556	*0.556*
Ours ($\tau = 0.80$)	*0.627*	0.558	0.554
Ours ($\tau = 0.95$)	**0.650**	*0.563*	**0.561**

Document Retrieval Layer. The results for the Document Retrieval Layer are expressed in terms of the Mean Reciprocal Rank (MRR) of the text fragment with the highest ROUGE-L/BLEURT score among the retrieved conclusions. For completeness, we report the performances here, despite them being strongly dependent on the embedding model employed for vectorizing the functional sections, and the index adopted by the DBMS for semantic search.

On the IT dataset, we obtain $MRR_{ROUGE-L} = 0.556$ and $MRR_{BLEURT} = 0.567$, meaning that the retrieved conclusion with the highest ROUGE-L/BLEURT score is, on average, between the first and the second position in the ranking, making it highly visible to the user. On the US dataset, instead, we get $MRR_{ROUGE-L} = 0.357$ and $MRR_{BLEURT} = 0.517$, a slight decrease in performance for the retrieved conclusion with the highest ROUGE-L score appearing in the second or the third position on average. As discussed in the original paper, "Conclusions" sections in US data substantially differ in content among themselves, sometimes presenting very short sentences or a completely different wording, resulting in a tough challenge even for domain experts. Since ROUGE-L scores relies on exact matches of common subsequences of words, the corresponding MRR is affected the most.

RAG Layer. Finally, in the RAG Layer, we wanted to compare the ability of two LLMs in generating conclusions, the quality of these conclusions with respect to retrieved ones, and the effect of the prompt augmentation on the generation process. The performances are measured in terms of ROUGE-L and BLEURT scores, whose distributions are displayed in Fig. 4(a) for US data and in Fig. 4(b) for IT data.

Our analysis reveals that:

- LLM performance: GPT-4o-Mini consistently outperformed the smaller Mistral-7B model across both datasets, producing text that was more similar to the ground truth. This is expected given its larger size and more advanced architecture. However, Mistral-7B's open-source nature offers crucial advantages for local deployment, ensuring data privacy, and could be a valid alternative in a security-centered scenario.

[8] https://huggingface.co/nlpaueb/legal-bert-base-uncased
[9] https://huggingface.co/dlicari/Italian-Legal-BERT
[10] https://www.trychroma.com
[11] https://huggingface.co/ccdv/lsg-bart-base-4096
[12] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

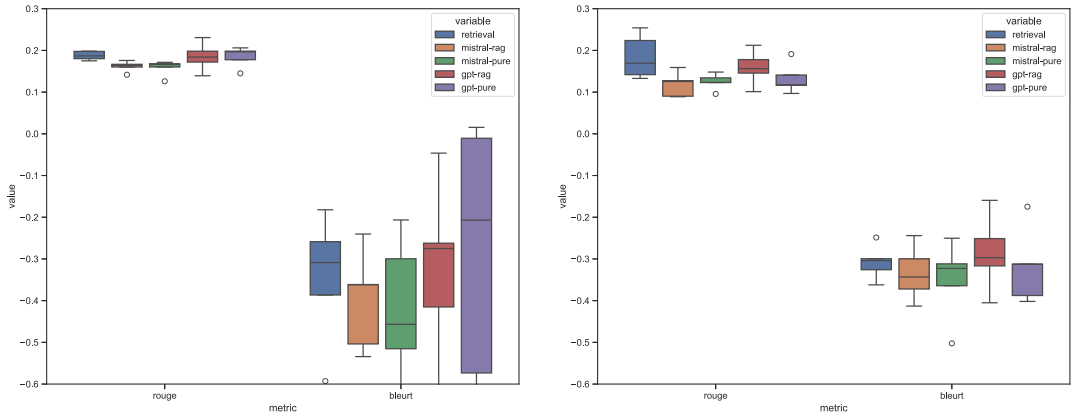

Fig. 4. ROUGE-L and BLEURT scores of the conclusions generated by Mistral-7B (with RAG in orange, otherwise in green) and GPT-4o-Mini (with RAG in red, otherwise in purple) compared to the scores of the best retrieval results (in blue), for US data (a) and IT data (b). Best visualized in color.

- Generated and retrieved section quality: the text generated by GPT-4o-Mini (with RAG) was often of comparable or higher quality than the best-retrieved example from the dataset. This suggests that the RAG model can synthesize information to create a suggestion that is more tailored to the current case than any single existing precedent.
- Impact of RAG: The addition of retrieved context (RAG) significantly improved generation quality on the IT dataset, especially for GPT-4o-Mini. On the more varied US dataset, the impact was more ambiguous, occasionally increasing variance. This suggests that the effectiveness of RAG is tied to the consistency of the underlying dataset.

In summary, our comprehensive evaluation provides clear answers to the three research questions that guided our investigation. Regarding document segmentation, our findings show that the architecture can effectively and robustly identify functional sections of legal judgments, demonstrating adaptability even when faced with different languages and annotation schemes. For document retrieval, the system proved effective, consistently placing the most relevant precedent sections at or near the top of the retrieved list, a critical feature to ensure practical usability. Finally, and most importantly, our analysis of the RAG Layer confirmed its value as a drafting assistant. The system generates text that is not only coherent and legally salient but, in many cases, offers a more tailored and useful starting point than any single retrieved example. Taken together, these results validate the JusBuild architecture as a practical and effective tool, capable of providing meaningful, AI-powered support in the complex task of drafting judicial decisions.

7. Concluding remarks

In this paper, we introduced JusBuild, a comprehensive architecture for legal document building which integrates advanced NLP techniques with human-centered design principles. JusBuild addresses the specific challenge of drafting operative sections of judicial rulings, distinguishing itself from general legal question-answering systems as well as previous attempts at creating a system tailored to specific parts of a legal document. Second, JusBuild embodies a "human-in-the-loop" design philosophy that explicitly prioritizes the preservation of judicial autonomy throughout the document building process, ensuring that AI enhancement never compromises the fundamental human authority in legal decision-making and positioning itself as a technological bridge that assists the evolution of modern legal systems rather than disrupting established judicial processes.

JusBuild effectively employs CRF models to segment legal precedents into functionally meaningful sections. Then, the system leverages vector database technology to efficiently perform semantic searches across vectorial representations of segmented legal content, dramatically improving retrieval accuracy and contextual relevance over keyword-based approaches. Most significantly, JusBuild provides intelligent, contextually aware suggestions by combining retrieved precedent sections with newly AI-generated section content produced through LLMs and RAG, thus creating a hybrid approach that maximizes both precedential grounding and adaptive reasoning capabilities.

JusBuild robustness and adaptability have been validated across diverse legal contexts through testing on real-world datasets that vary in language, section schema, and judicial subject matter. This cross-jurisdictional validation demonstrates JusBuild's potential to support legal document building across different legal systems and languages, establishing its significance as a versatile tool for global legal practice while respecting the procedural specificities of individual jurisdictions.

JusBuild represents a paradigm shift toward intelligent and contextually aware legal document assistance that bridges the gap between technological advancement and judicial tradition. As legal systems worldwide continue their digital transformation, JusBuild demonstrates how AI can serve as a collaborative partner in the evolution of legal practice, supporting rather than supplanting the essential human expertise, by concretely showing how to address the need of balancing automation provided by technological features with preservation of judge autonomy and human rights protection, as required by relevant European regulations on AI and data protection.

Future developments promise even greater impact through targeted enhancements that build upon JusBuild's innovations. While domain-specific fine-tuned embedding models already exist for legal applications, text generation currently relies on general-purpose models that present significant opportunities for specialized optimization. Fine-tuning these generative models using legal practitioner feedback as additional training data could substantially improve architectural performance, creating increasingly sophisticated and contextually appropriate content suggestions that better reflect the nuances of judicial reasoning. Additionally, exploring specialized prompting techniques tailored to specific document types and legal contexts offers potential for further customization and effectiveness improvements that could enhance the system's adaptability across different judicial frameworks. The automated generation of textual suggestions for an active section, may exert influence upon judicial decision-making processes through the introduction of framing effects (by presenting suggestions in a certain order), availability biases (by presenting some case types more frequently than others), and inappropriate analogical reasoning or overgeneralization. Exploring these issues as a future interdisciplinary research goal together with legal practitioners would contribute to mitigate or overcome these risks to further enhance the practical usability of JusBuild.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: All authors report financial support was provided by European Union and Italian Ministry of University and Research (MUR). No other known competing financial interests or personal relationships could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by MUR 118/2023 and by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the Italian MUR. Neither the European Union nor the Italian MUR can be held responsible for them.

Data availability

The authors do not have permission to share data.

References

[1] Savelka J, Ashley KD. Segmenting US court decisions into functional and issue specific parts. In: JURIX. 2018, p. 111–20.
[2] Ferrara A, Picascia S, Riva D. Few-shot legal text segmentation via rewiring conditional random fields: A preliminary study. In: International conference on conceptual modeling. Springer; 2023, p. 141–50.
[3] Gordon T. A theory construction approach to legal document assembly. In: Preproceedings of the third international conference on logic, informatics, and law. Citeseer; 1989, p. 485–98.
[4] Branting LK. An issue-oriented approach to judicial document assembly. In: Proceedings of the 4th international conference on artificial intelligence and law. 1993, p. 228–35.
[5] Evans DB. Artificial intelligence and document assembly. Law Pr Mgmt. 1990;16:18.
[6] Palmirani M, Governatori G, Rotolo A, Tabet S, Boley H, Paschke A. LegalRuleML: XML-based rules and norms. RuleML Am 2011;7018:298–312.
[7] Marković M, Gostojić S. Knowledge-based legal document assembly. 2020, arXiv preprint arXiv:2009.06611.
[8] Robaldo L, Bartolini C, Lenzini G. The DAPRECO knowledge base: representing the GDPR in LegalRuleML. In: Proceedings of the 12th language resources and evaluation conference. 2020, p. 5688–97.
[9] Marković M, Gostojić S. Legal document assembly system for introducing law students with legal drafting. Artif Intell Law 2022;31(4):829–63. http://dx.doi.org/10.1007/s10506-022-09339-2.
[10] Castano S, Ferrara A, Montanelli S, Picascia S, Riva D. A knowlegde-based service architecture for legal document building. In: 2nd international workshop on knowledge management and process mining for law, vol. 3637, Sherbrooke, Quebec, Canada; 2023, p. 1–15.
[11] Glavaš G, Nanni F, Ponzetto SP. Unsupervised text segmentation using semantic relatedness graphs. In: Gardent C, Bernardi R, Titov I, editors. Proceedings of the fifth joint conference on lexical and computational semantics. Berlin, Germany: Association for Computational Linguistics; 2016, p. 125–30. http://dx.doi.org/10.18653/v1/S16-2016, URL https://aclanthology.org/S16-2016.
[12] Koshorek O, Cohen A, Mor N, Rotman M, Berant J. Text segmentation as a supervised learning task. In: Walker M, Ji H, Stent A, editors. Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 2 (short papers). New Orleans, Louisiana: Association for Computational Linguistics; 2018, p. 469–73. http://dx.doi.org/10.18653/v1/N18-2075, URL https://aclanthology.org/N18-2075.
[13] Glavaš G, Ganesh A, Somasundaran S. Training and domain adaptation for supervised text segmentation. In: Burstein J, Horbach A, Kochmar E, Laarmann-Quante R, Leacock C, Madnani N, Pilán I, Yannakoudakis H, Zesch T, editors. Proceedings of the 16th workshop on innovative use of NLP for building educational applications. Online: Association for Computational Linguistics; 2021, p. 110–6, URL https://aclanthology.org/2021.bea-1.11.
[14] Solbiati A, Heffernan K, Damaskinos G, Poddar S, Modi S, Cali J. Unsupervised topic segmentation of meetings with BERT embeddings. 2021, arXiv:2106.12978. URL https://arxiv.org/abs/2106.12978.
[15] Aumiller D, Almasian S, Lackner S, Gertz M. Structural text segmentation of legal documents. In: Proceedings of the eighteenth international conference on artificial intelligence and law. New York, NY, USA: Association for Computing Machinery; 2021, p. 2–11. http://dx.doi.org/10.1145/3462757.3466085.
[16] Bhattacharya P, Paul S, Ghosh K, Ghosh S, Wyner A. Identification of rhetorical roles of sentences in Indian legal judgments. 2019, arXiv:1911.05405. URL https://arxiv.org/abs/1911.05405.
[17] Sansone C, Sperlí G. Legal information retrieval systems: State-of-the-art and open issues. Inf Syst 2022;106:101967. http://dx.doi.org/10.1016/j.is.2021.101967, URL https://www.sciencedirect.com/science/article/pii/S0306437921001551.
[18] Mok WY, Mok JR. Legal machine-learning analysis: First steps towards a.I. Assisted legal research. In: Proceedings of the seventeenth international conference on artificial intelligence and law. New York, NY, USA: Association for Computing Machinery; 2019, p. 266–7. http://dx.doi.org/10.1145/3322640.3326737.
[19] Klein MC, Van Steenbergen W, Uijttenbroek EM, Lodder AR, van Harmelen F. Thesaurus-based retrieval of case law.. Frontiers Artificial Intelligence Appl 2006;152:61.
[20] Castano S, Ferrara A, Falduti M, Montanelli S. Crime knowledge extraction: An ontology-driven approach for detecting abstract terms in case law decisions. In: Proc. of the 17th int conference on artificial intelligence and law. New York, NY, USA: ACM; 2019, p. 179–83. http://dx.doi.org/10.1145/3322640.3326730.
[21] Ferrara A, Picascia S, Riva D. Context-aware knowledge extraction from legal documents through zero-shot classification. In: Proc. of the 1st ER int. workshop on digital justice, digital law, and conceptual modeling. Hyderabad, India; 2022, p. 81–90.
[22] Elnaggar A, Otto R, Matthes F. Deep learning for named-entity linking with transfer learning for legal documents. In: Proceedings of the 2018 artificial intelligence and cloud computing conference. New York, NY, USA: Association for Computing Machinery; 2018, p. 23–8. http://dx.doi.org/10.1145/3299819.3299846.
[23] Shao Y, Mao J, Liu Y, Ma W, Satoh K, Zhang M, Ma S. BERT-PLI: Modeling paragraph-level interactions for legal case retrieval. In: Bessiere C, editor. Proceedings of the twenty-ninth international joint conference on artificial intelligence. International Joint Conferences on Artificial Intelligence Organization; 2020, p. 3501–7. http://dx.doi.org/10.24963/ijcai.2020/484, Main track.
[24] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, tau Yih W, Rocktäschel T, Riedel S, Kiela D. Retrieval-Augmented Generation for knowledge-intensive NLP tasks. 2021, arXiv:2005.11401. URL https://arxiv.org/abs/2005.11401.
[25] Gao Y, Xiong Y, Gao X, Jia K, Pan J, Bi Y, Dai Y, Sun J, Wang M, Wang H. Retrieval-Augmented Generation for large language models: A survey. 2024, arXiv:2312.10997. URL https://arxiv.org/abs/2312.10997.
[26] Wiratunga N, Abeyratne R, Jayawardena L, Martin K, Massie S, Nkisi-Orji I, Weerasinghe R, Liret A, Fleisch B. CBR-RAG: Case-based reasoning for retrieval augmented generation in LLMs for legal question answering. In: Recio-Garcia JA, Orozco-del Castillo MG, Bridge D, editors. Case-based reasoning research and development. Cham: Springer Nature Switzerland; 2024, p. 445–60.
[27] Yang R. CaseGPT: a case reasoning framework based on language models and retrieval-augmented generation. 2024, arXiv:2407.07913. URL https://arxiv.org/abs/2407.07913.
[28] Cui J, Ning M, Li Z, Chen B, Yan Y, Li H, Ling B, Tian Y, Yuan L. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model. 2024, arXiv:2306.16092. URL https://arxiv.org/abs/2306.16092.
[29] Louis A, van Dijck G, Spanakis G. Interpretable long-form legal question answering with retrieval-augmented large language models. In: Proceedings of the AAAI Conference on Artificial Intelligence, In: Proceedings of the AAAI Conference on Artificial Intelligence, 2024;vol. 38(20):22266–75. http://dx.doi.org/10.1609/aaai.v38i20.30232.URL https://ojs.aaai.org/index.php/AAAI/article/view/30232,
[30] Chouhan A, Gertz M. LexDrafter: Terminology drafting for legislative documents using retrieval augmented generation. In: Calzolari N, Kan M-Y, Hoste V, Lenci A, Sakti S, Xue N, editors. Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation. Torino, Italia: ELRA and ICCL; 2024, p. 10448–58, URL https://aclanthology.org/2024.lrec-main.913.
[31] Hou AB, Weller O, Qin G, Yang E, Lawrie D, Holzenberger N, Blair-Stanek A, Durme BV. CLERC: A dataset for legal case retrieval and retrieval-augmented analysis generation. 2024, arXiv:2406.17186. URL https://arxiv.org/abs/2406.17186.
[32] Castignone S, Guastini R, Tarello G. Introduzione teorica allo studio del diritto. Genoa: ECIG; 1978.
[33] Tarello G. L'interpretazione della legge. Milan: Giuffré; 1980.
[34] Tarello G. Il realismo giuridico americano. Rome: RomaTrE- Press; 2023.
[35] Ehrlich E. Grundlegung der Soziologie des Rechts. Berlin: Duncker & Humblot; 1913.
[36] Castignone S, Faralli C, Ripoli M. Il diritto come profezia. Il realismo americano: antologia di scritti. Turin: Giappichelli; 2002.
[37] Frank J. Law and the modern mind. Transaction Publishers; 1930.

[38] Oswald M. Algorithm-assisted decision-making in the public sector: framing the issues using administrative law rules governing discretionary power. Philos Trans R Soc A: Math Phys Eng Sci 2018;376(2128):20170359.
[39] Lafferty J, McCallum A, Pereira F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML '01: proceedings of the eighteenth international conference on machine learning. 2001, p. 282–28.
[40] Condevaux C, Harispe S. Lsg attention: Extrapolation of pretrained transformers to long sequences. In: Pacific-Asia conference on knowledge discovery and data mining. Springer; 2023, p. 443–54.
[41] Zanoli E, Barbini M, Riva D, Picascia S, Furiosi E, D'Ancona S, Chesi C, et al. Annotators-in-the-loop: testing a novel annotation procedure on Italian case law. In: Proceedings of the 17th linguistic annotation workshop. Association for Computational Linguistics; 2023, p. 118–28.
[42] Pinotti G, Santosuosso A, Fazio F. A rule 74 for Italian judges and lawyers. In: Advances in conceptual modeling: ER 2022 workshops, CMLS, empER, and JUSmOD, hyderabad, India, October 17–20, 2022, proceedings. Springer; 2023, p. 112–21.
[43] Lin C-Y, Och FJ. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In: Proceedings of the 42nd annual meeting of the association for computational linguistics. 2004, p. 605–12.
[44] Sellam T, Das D, Parikh AP. BLEURT: Learning robust metrics for text generation. 2020, arXiv preprint arXiv:2004.04696.
[45] Devlin J, Chang M, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein J, Doran C, Solorio T, editors. 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). Association for Computational Linguistics; 2019, p. 4171–86. http://dx.doi.org/10.18653/V1/N19-1423.
[46] Zhang H, Gong Y, Yan Y, Duan N, Xu J, Wang J, Gong M, Zhou M. Pretraining-based natural language generation for text summarization. 2019, CoRR abs/1902.09243, arXiv:1902.09243. URL http://arxiv.org/abs/1902.09243.
[47] An C, Feng J, Lv K, Kong L, Qiu X, Huang X. CoNT: Contrastive neural text generation. In: Koyejo S, Mohamed S, Agarwal A, Belgrave D, Cho K, Oh A, editors. In: Advances in neural information processing systems, vol. 35, Curran Associates, Inc.; 2022, p. 2197–210, URL https://proceedings.neurips.cc/paper_files/paper/2022/file/0f5fcf4bff73a3537e0813a38f0d3f76-Paper-Conference.pdf.
[48] Moosavi NS, Rücklé A, Roth D, Gurevych I. SciGen: a dataset for reasoning-aware text generation from scientific tables. In: Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 2). 2021, URL https://openreview.net/forum?id=Jul-uX7EV_I.
[49] Malkov YA, Yashunin DA. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Trans Pattern Anal Mach Intell 2018;42(4):824–36.
[50] Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, Casas Ddl, Bressand F, Lengyel G, Lample G, Saulnier L, et al. Mistral 7B. 2023, arXiv preprint arXiv:2310.06825.