A Knowledge-Based Service Architecture for Legal Document Building

Silvana Castano[1], Alfio Ferrara[1], Stefano Montanelli[1], Sergio Picascia[1] and Davide Riva[1]

[1]*Università degli Studi di Milano*
Department of Computer Science
Via Celoria, 18 - 20133 Milano, Italy

Abstract

In this paper, we propose a knowledge-based service architecture for legal document building based on Natural Language Processing and learning techniques, to semantically analyze a database of ingested legal documents and propose the most prominent and pertinent textual suggestions for new document composition. After describing the proposed NLP services for knowledge extraction and textual suggestion selection and proposition, we describe the application of proposed document builder architecture by considering a case study of Italian civil judgements.

Keywords

Digital Justice, Knowledge Extraction, Legal Concept Graph

1. Introduction

Legal documents constantly produced by Parliaments, Courts, and other institutional bodies constitute a prominent source of information and knowledge not only for legal actors, like judges or lawyers, but also for general subjects, like citizens or private and public organizations. To improve both efficiency and effectiveness of courthouses and legal record offices, and to foster digital justice, a significant effort is being devoted in almost all countries to digital transformation projects, by developing legal information systems and modular architectures providing a variety of services for acquisition, management, classification, exploration, and retrieval of legal documents [1]. A main issue fostering digital transformation and courthouses efficiency, is related to the availability of tools to support legal actors in the complex process of producing a new document for the case at hand, by making available databases of previous documentation as well as workflow management services for the different stages of the legal proceedings [2]. In particular, advanced legal document building environments are required, to assist legal actors in the production of a new document, like judgements or lawyer acts, by relying on predefined document templates and knowledge-based services to properly extract

KM4LAW: 2nd international workshop on Knowledge Management and Process Mining for Law, July 17–20, 2023, Sherbrooke, Québec, Canada

silvana.castano@unimi.it (S. Castano); alfio.ferrara@unimi.it (A. Ferrara); stefano.montanelli@unimi.it (S. Montanelli); sergio.picascia@unimi.it (S. Picascia); davide.riva1@unimi.it (D. Riva)

0000-0002-3826-2407 (S. Castano); 0000-0002-4991-4984 (A. Ferrara); 0000-0002-6594-6644 (S. Montanelli); 0000-0001-6863-0082 (S. Picascia); 0009-0003-9681-9423 (D. Riva)

CEUR Workshop Proceedings (CEUR-WS.org)

useful information available in previously produced legal documents to propose in form of suggestions. The availability of predefined document templates [3] and rules [4] promotes a disciplined structuring of legal documents, which will be eventually ingested in the document repository for subsequent processing. The availability of a legal document repository where ingested documents have a very homogeneous structure can, in turn, facilitate the application of AI techniques, for semantic processing and classification of document contents.

In this paper, we propose a knowledge-based service architecture for legal document building based on Natural Language Processing and Zero-Shot Learning techniques, to semantically analyze a database of ingested legal documents and propose the most prominent and pertinent textual suggestions for the composition of a new document. In particular, the document building process relies on a predefined document template organized in sections, according to a segmentation schema, and on a *knowledge-extraction service* and a *suggestion-extraction service* to enforce the document building process. The *knowledge-extraction service* is responsible for i) mining a set of featuring concepts that provide a topic-oriented description of textual contents of ingested documents and ii) implementing a fine-grained semantic classification of textual contents, where each concept is connected to the document portions from which the concept emerged. The *suggestion-extraction service* is responsible for i) *Search-by-Content*, to retrieve text portions of ingested documents that are most pertinent to the current text of the document section under preparation; ii) *Search-by-Concept*, to retrieve text portions of ingested documents that are most pertinent to one or more concepts, and possibly document section(s) of interest, specified by the legal actor, that is, text portions that contain terminological occurrences of the considered concept(s), coming from specified section(s), if any. The proposed service architecture supports the legal document building process according to a "human-in-the-loop approach", where the document author, namely the legal actor, drafts the document under preparation following the document template, by actively working on the text portions provided by the suggestion service.

The paper is organized as follows. In Section 2 we introduce the proposed service architecture for a legal document builder. Section 3 describes the knowledge-extraction service, while Section 4 illustrates the suggestion-extraction service. In Section 5 we present a case study applying the architecture on a corpus of Italian case law decisions in the framework of the *NGUPP* project. Section 6 is devoted to related work. Finally, Section 7 provides the final remarks.

2. Legal document builder: the proposed architecture

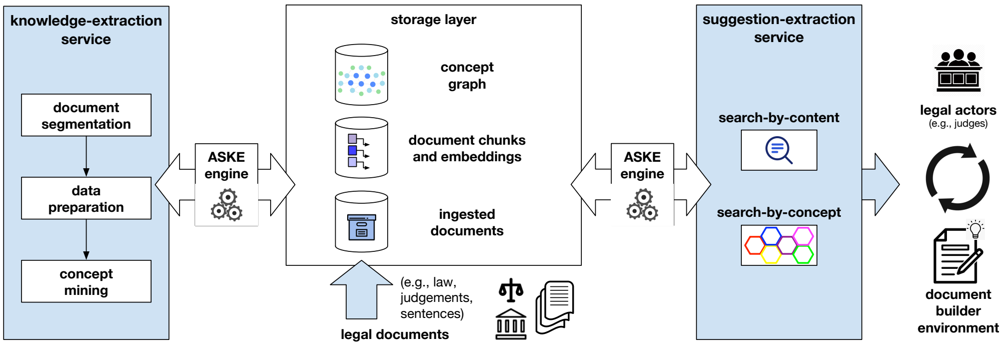
Document building, sometimes referred to as *document assembly*, is a process that aims at producing a textual document following a predefined schema with the support of digital, automated tools. The task, even when entirely performed by humans, can be thought of as a sequence of three phases: (1) definition of the document format and structure; (2) content draft of each part or section; (3) 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 may be supported by solutions ranging from error detection to automated rephrasing and text generation.

Our proposed architecture is designed to support experts in producing documents specialized domains that require (or would benefit from) a well-defined document structure, such as the legal domain, which is the context we consider in this paper. A main requirement in designing a document builder architecture for the legal domain is to preserve the importance of the legal actor role and work with capability to provide automated support while preserving the uniqueness/autonomy of the assessment and judgment capability of legal actors in document writing. As a consequence, a fully automated approach is not viable nor desirable, and interactive tool environments should be provided, to keep the legal actor "in-the-loop" and to reduce also the risk of excessive standardization of legal documents, which is a requirement too, given the value and purposes of legal documents.

The architecture provides a set of services to support the document assembly process by relying on predefined document template(s). This way, document building promotes a disciplined approach to document generation, by enhancing the readability, homogeneity, and quality of produced documentation, very important also for subsequent analysis and knowledge extraction purposes.

The proposed service architecture for legal document building is shown in Figure 1. The architecture aims to support *legal actors* like judges and judicial officers in the preparation of official legal documents. In particular, the legal actor, namely the document author, starts working by selecting a template among those available and she/he is supported in building the document by receiving suggestions in the form of pertinent chunks of text that can be reused "as-is" or adjusted/edited for insertion in the document under preparation. Suggestions are extracted from a corpus of legal documents (e.g., judgements, case-law decisions) that have been previously ingested and processed. To this end, a storage layer is defined to maintain i) the document database for the raw *ingested documents* and corresponding texts; ii) the *document chunks* and related embeddings extracted from documents for classification; iii) the *concept graph* to store entries for the concepts extracted from documents.

Figure 1: The proposed service architecture.



Two main service pipelines are defined: the *knowledge-extraction service* and the *suggestion-extraction service*. In the knowledge-extraction service, appropriate components are defined to exploit the ingested documents for mining a set of featuring concepts that provide a topic-oriented description of underlying textual contents. The concepts extracted from the documents are organized in a graph, where a pair of similar concepts is linked by an edge. Each concept is also connected to the document portions from which the concept emerged, meaning that we can retrieve the pertinent document segments where a certain concept occurs. For knowledge extraction, the involved components are *document segmentation* and *data preparation*, and *concept mining*.

The *suggestion-extraction service* enforces two modalities to detect pertinent chunks to propose to the user for document drafting: i) *Search-by-Content*, which retrieves similar chunks as suggestions by considering the current text of the document section under preparation; ii) *Search-by-Concept*, which is based on a concept specified by the legal actor to retrieve and provide as suggestions those phrases that contain terminological occurrences of the considered concept.

In the following sections, we describe each service in more detail.

3. The knowledge-extraction service

Ingested legal documents are submitted to a knowledge-extraction pipeline based on document annotation/segmentation, and fine-grained indexing and classification techniques.

3.1. Document segmentation

Document segmentation is concerned with the identification of relevant sections of a document which are exploited to construct new documents and to filter suggestions for the document drafting. The idea is that the legal actor receives suggestions extracted from the same, or strictly related, document section that is currently under construction.

Document segmentation is a type of document annotation that consists in dividing a document into parts (known as *segments*) that "display local coherence" [5], eventually assigning a label to each of them. Since in our case we focus only on textual data, such parts may be contiguous sequences of words, clauses or sentences. Local coherence is here interpreted from a functional point of view, i.e. a text segment plays a single and indivisible function in the structure of the document, e.g. introduction, argumentation, conclusion, and others.

The first problem to solve is therefore the definition of a segmentation schema comprising a set of functional segments and the descriptions of the respective text characteristics, which serve as guidelines for annotation. Binding rules over the annotation output can be specified by an additional set of axioms, including for instance a lower and/or upper bound on the number of segments of a certain type. In most cases, definition of segment functions and characteristics and axioms over them requires expert knowledge of the domain at hand, and shall be general enough to encompass the variety of documents in the corpus of interest.

The second problem is the actual annotation of documents, complying with the schema. Given a segmentation schema and a corpus to annotate, the annotation can be performed either manually (by human annotators), automatically (by rule-based or machine learning techniques), or by a hybrid approach.

Manual Segmentation. In the manual case, the annotation is performed by a group of human annotators, generally, experts of the domain, who split each document into segments and assign a label from the segmentation schema to each segment using a digital text annotation tool. Leveraging expert knowledge can benefit annotation accuracy, especially in case of complex documents or domain corpora like in the legal domain, but it hinders scalability, since manual annotation is notoriously a labour-intensive, time-consuming activity. Moreover, despite increasing accuracy, expert knowledge doesn't ensure agreement among annotators, which is often necessary to define a unique and homogeneous ground truth.

Automated Segmentation. Automated systems for text segmentation typically rely on either rule-based or machine learning models. Such models ensure two main advantages with respect to manual annotation: they can achieve higher scalability, making it possible to process large corpora, and, if machine learning models are employed, text characteristics can be learned automatically instead of being a-priori defined. The main disadvantage is lower accuracy, especially on complex documents. The problem may intensify in case of totally automated systems, which rely on pre-trained models only, in the absence of any manually annotated data.

Hybrid Segmentation. By hybrid segmentation we indicate a text segmentation system that receives as input a limited corpus of manually annotated documents and, training and validating an automated system on such data, extends the annotation to other documents.

In Section 5, we discuss legal document segmentation activity we performed for prototyping a first release of a legal document builder in the framework of an ongoing research project on digital justice in Italy. In this context, legal document segmentation has been manually performed by legal experts of the project on a dataset case study of judgements. Since rule-based models usually fall short in generalization, we deem machine learning models preferable to process large and possibly heterogeneous legal corpora. We plan experimentation of different classifiers as discussed in concluding remarks.

3.2. Data preparation

Data preparation is concerned with the tokenization of ingested documents with the goal to split them into document chunks. A *document chunk* represents the text unit to consider for classification that can be associated with a concept. Document chunks associated with concepts represent the suggestions that can be provided to the author while building a document. We stress that the size of the document chunk should be large enough, so that the context can be captured, but not too much extended to avoid segments that are long to read and potentially noisy due to the presence of multiple concepts. In this paper, we choose to tokenize documents by defining a chunk for few sentence/phrase detected in a document, up to a maximum size of 128 tokens[1]. This is particularly appropriate for legal actors that are typically interested in retrieving precise suggestions in which a given concept of interest appears and can be rapidly read/assimilated.

As a further preparation step, the terms appearing in document chunks are lemmatized and a vector-based representation of each document chunk is finally built. The use of embedding techniques to represent chunks allows to map the document contents on a semantic vector space where the similarity of two chunks can be measured by comparing the corresponding vector representations through a similarity metric (e.g., cosine similarity). For embedding construction, Sentence-BERT [6], a modification of the original BERT model based on siamese and triplets networks, is employed to derive a semantically meaningful embedding for a given

[1] The size of the document chunk is determined by the maximum number of tokens that can be processed by the considered embedding model.

sentence/phrase. As such, a document chunk k is associated with a set of terms W_k therein contained. Any term is described as $w = (w_l, w_d, \bar{w})$, where w_l is the label of the term (i.e., the lemma), w_d is a description of the term meaning taken from a reference dictionary/vocabulary (e.g., WordNet), and $\bar{w}$ is the corresponding vector-based representation according to Sentence-BERT, respectively. A document chunk k has the form $k = (s_d, k_d, \bar{k})$, where s_d is the section of the document where the chunk occurs, k_d is the original textual content of the chunk, and $\bar{k}$ is the corresponding vector-based representation calculated as the mean of term vectors $\bar{w}$ with $w \in W_k$. Embedding models have the capability to represent and compare the meaning of entire text blocks like document chunks. On such a target, context-aware embedding models fine-tuned on document similarity tasks, like Sentence-BERT, are appropriate. In the legal field, the phrase structure can be highly articulated, and some common terms can have a precise technical meaning when used in a court (e.g., citation, clemency, designation). Sentence-BERT can handle such a kind of situations, which may strongly deviate with respect to everyday conversations.

3.3. Concept mining

Our solution to concept mining is called ASKE and it is based on zero-shot learning techniques and context-aware embedding models to enforce concept extraction [7].

Zero-shot learning is an unsupervised classification technique, characterized by the capability to enforce classification without requiring any pre-existing annotation of the considered documents. Initially, a *seed knowledge* is defined as a set of textual descriptions, each one featuring a concept of interest, namely a *seed concept*, to consider for classification. Typically, for a seed concept, a basic, gross-grained description is provided as a short text (e.g., one or two phrases) or a list of keywords. As an example, for a seed concept about banking contract, a corresponding textual description used for embedding is bank deposit, safe deposit box, bank credit opening, bank advance, bank account, bank discount. Further concepts are derived from seed ones during the extraction process, and they usually provide a more fine-grained description of the concept instances occurring in the document chunks. A concept c, either seed or derived, is defined as a pair $c = (c_l, \bar{c})$, where c_l is a label featuring the meaning of the concept expressed in a synthetic and human-understandable way, and $\bar{c}$ is a vector-based concept representation. Each concept c is initially associated with the set of terms W_c extracted from the textual description of c. The vector concept $\bar{c}$ is built as the mean of the vectors of all the terms in W_c. Finally, the label c_l corresponds to the label w_l of the term $w \in W_c$, whose vector representation $\bar{w}$ is closest to the concept vector $\bar{c}$. Concept extraction is defined as a progressive, iterative process articulated in the following three steps:

Zero-shot classification. Given a set of concepts (i.e., the seed concepts at the beginning of the process), the document chunks are classified through zero-shot learning. A similarity measure σ, e.g. cosine similarity, is calculated over any pair of embeddings between chunks and concepts. A document chunk k is classified with the concept c when the similarity value satisfies $\sigma(\bar{k}, \bar{c}) \geq \alpha$, with α defined as a similarity threshold configured in the system. The value of α is empirically determined according to experimental results. In this paper, the value $\alpha = 0.3$ is employed in the proposed case-studies and experiments.

Terminology enrichment. Given a document chunk k classified with the concept c, the terms in W_k are exploited for enriching the term set W_c. The idea is that the initial description of the concept c can become more detailed if we add terminology taken from chunks that are pertinent (i.e., classified) with c. This is done by summing, for each $w \in W_k$, similarities $\sigma(\overline{w}, \overline{c})$ and $\sigma(\overline{w}, \overline{K_c})$, where $\overline{K_c}$ denotes the average embedding of document chunks classified with c. Terms $w \in W_k$ satisfying a system-defined β similarity threshold are inserted in W_c.

Concept derivation. By enriching the term set W_c, it is possible that more fine-grained concepts emerge from c, and they can be generated as new concepts. The discovery of possible new concepts emerging from c is enforced by clustering the embedding vectors $\bar{w}$ of terms in W_c. The Affinity Propagation (AP) algorithm is adopted to this end, since it allows to detect the emergence of sub-groups of similar terms within W_c, without requiring to "a-priori define" the number of clusters to generate. A new concept c' is created for each cluster returned by AP on the terms W_c of a concept c. A link is defined between a concept c' and c to denote that c' is derived from c and they are somehow similar/related in content. The concept c is then updated since the terms in W_c can be changed due to enrichment. As a consequence, c_l and $\bar{c}$ are re-calculated.

The set of concepts obtained after derivation can trigger the execution of a new cycle based on the above three steps. New derived concepts can contribute to improve the classification of chunks with more fine-grained concepts. Further new concepts can be also discovered through a new execution of enrichment and derivation on the basis of a refined classification result. As such, concept extraction is characterized by a predefined endpoint condition based on a *termination threshold.* When the number of new concepts created in the derivation step is lower than the threshold, the concept extraction process is concluded. A concept graph providing a topic-based description of the underlying document corpus is finally stored.

4. Suggestion-extraction for document drafting

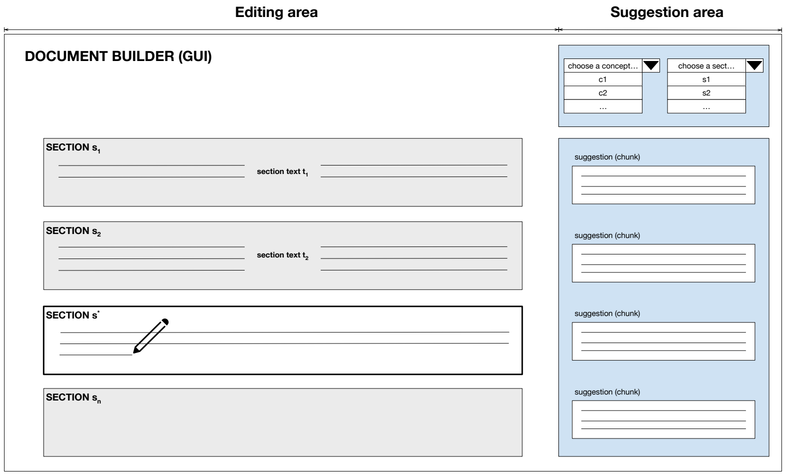
This service supports the legal document building process, by implementing a human-in-the-loop approach, where the document author, namely the legal actor, has to draft a document under preparation and she/he receives suggestions extracted by the ASKE engine according to the results of the knowledge-extraction service.

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 author can retrieve pertinent chunks to be re-used and optionally changed in the currently-edited section (see Figure 2). The author is mainly focused on the editing area with the aim to write and create the document. The structure of the document follows a template that the author has to choose within a predefined set. A document template is articulated in a list of sections $s_1, \dots, s_n$. In the following, we call d^* the document under editing (i.e., the *active document*), and s^* the section on which the author is currently focused for drafting (i.e., the *active section*).

Two modalities are defined to provide suggestions to the author during document drafting:

- *Search-by-Content*, which enforces similarity-based retrieval of text suggestions for the section s^* by exploiting the chunks of ingested documents;
- *Search-by-Concept*, which enforces concept-based retrieval of text suggestions for the section s^* by exploiting the concept graph.

Figure 2: The Document Builder Environment.



4.1. Search-by-Content

In the Search-by-Content modality, suggestions are represented by document chunks belonging to ingested documents that are considered most similar to the active one; the degree of similarity is computed taking into consideration document chunks belonging to a section preceding s^*. The following steps are executed to determine the most pertinent suggestions for an active section s^* with text t^*.

Detection of similar documents on previous sections. Call $s_1, \dots, s_k$ the sections of the active document d^* that precede the active section s^*. The ASKE engine is invoked to determine the ingested documents that contain similar chunks with respect to the texts $t_1, \dots, t_k$ of sections $s_1, \dots, s_k$, respectively. For each section s_i of the active document d^* with $i \leq k$, the corresponding section content t_i is passed to the ASKE engine. The ASKE embedding model, namely the model used for knowledge extraction, is queried with the aim of extracting a vector representation $\vec{t_i}$ of t_i[2]. ASKE employs the cosine similarity function to compare $\vec{t_i}$ against the document chunks of the s_i section belonging to any ingested document d. For each document d, a similarity degree σ_d is defined by summing all the similarity values provided by the matching chunks k_d in s_i. For a section s_i, a set of similar documents D_i is returned as a result, where the similarity degree σ_d of a document $d \in D_i$ is over a prefixed system-designed threshold. An overall set of similar documents $D = \bigcap_{i=1}^{i=k} D_i$ is finally returned to use as input for the next step.

Detection of pertinent chunks on the active section. Given the active section s^*, we retrieve a set of pertinent suggestions among the chunks of the similar documents in D. As a default behavior, the chunks of section s^* in the documents of D are retrieved as suggestions to support the author in drafting the active section. Moreover, to obtain a refined set of suggestions, the author can start drafting the active section s^* by inserting an initial section content t^*. In this case, ASKE is invoked to derive the vector representation $\vec{t^*}$ of t^*. The embeddings of chunks

[2] When the section content t_i exceeds the size of a document chunk, a tokenization mechanism is employed to spit t_i. For the sake of clarity, in the paper, we consider t_i as a single textual element with a corresponding vector-based representation $\vec{t_i}$.

in documents of D and section s^* are compared against $\vec{t}^*$. A ranked list of similar chunks are returned as suggestions to show in the right-hand panel of Figure 2 (descending order).

4.2. Search-by-Concept

In the Search-by-Concept modality, suggestions represent the document chunks where an occurrence of a given concept of interest appears according to the classification results of ASKE.

In the suggestion area, the author can select a concept of interest c^* among a set of available ones, namely the set of concepts derived by ASKE during knowledge extraction from the ingested documents. The concept graph of ASKE is queried to retrieve the set of document chunks K_{c^*} classified as c^* that are then returned as a result.

A further filtering option is provided in the suggestion area to enable the author to choose a target section of interest s^*. When a target section is selected, only the chunks of K_{c^*} belonging to the section s^* in the ingested documents are finally shown as suggestions in the right-hand panel of Figure 2.

5. Application to the Italian legal domain

The proposed service architecture has been applied to a legal case study in framework of the *Next Generation UPP* Italian project, aimed at providing artificial intelligence and advanced information management techniques for digital transformation of Italian legal processes and digital law in general.

We firstly introduce the dataset employed in the case study, explaining document preparation and segmentation. Then, we describe the scenario in which a legal actor, i.e. the judge, makes use of the document builder functionalities in order to generate a new judgement, with examples of Search-by-Content and Search-by-Concept text suggestions. The case study has been conducted on documents written in Italian; examples reported in the paper have been translated into English for the purpose of understanding. Also, for the sake of brevity, we report an excerpt of the text suggestions.

5.1. Dataset

The dataset consists in a corpus of 50 Italian case law decisions, retrieved from 12 different Courts located in Northern Italy. All the documents concern first degree civil law judgments regarding the matter of unfair competition. Such kind of documents comes in PDF files of different formats, depending on the court emitting it; plain text has been extracted from these files in order to be used in the document segmentation and the data preparation processes. For document segmentation, our legal partners of the project provided a document schema defined on the basis of the Rules of the Court (March 2022). This set of rules has the function of giving a rigid and predetermined structure to the introductory acts of the parties and to the decision of the Court. In particular, for example, Rule 74 states that "any judgment of the European Court of Human Rights must contain some basic information (such as the names of the parties, agents, lawyers or advisers of the parties and a report of the procedure followed) followed by some well-defined steps of the decision: the facts of the case; a summary of the arguments of the parties; the legal reasons; operational provisions". Similar criteria are also established for the drafting of the appeals of lawyers. Based on Rule 74 recommendations and related literature, our law project partners developed a document segmentation schema for Italian juridical judgements/case law decisions [3], which has been enforced in our proposed architecture, for document segmentation and for document building. The segmentation schema comprises five different structural sections with the following corresponding labels (identified directly by the domain experts): *court and parties*, providing information of the court, the panel of judges and the parties in the trial; *background information*, relating to the proceedings of the trial and to the reconstruction of the facts; *claims and arguments*, claims made by the plaintiffs and counterclaims made by the defendants; *reasoning*, for each decision about an individual claim; *final decisions*, for each individual claim. For the purposes of the case study, document segmentation has been performed manually by legal domain experts involved in the project.

5.2. Search-by-Content

In the Search-by-Content scenario, the legal actor is required to provide to the system basic knowledge regarding the case at hand, filling at least a section s of the chosen document template, in order to receive suggestions on an active section s^*. These suggestions are retrieved from previous relevant case law decisions stored in the database of ingested documents and sorted based on the computed similarity scores. In particular, the legal actor could write down the sections s_1 and s_2 regarding *background information* and *claims and arguments* in order to receive suggestions for the *reasoning* section s^*. We exclude from the analysis the first section in the aforementioned annotation schema, *court and parties*, since it is auto-compiled by the software tools currently in use by legal actors and it includes general information that end up being not relevant for the search process.

As an example of document building, suppose to work on the generation of a new judgement regarding the `counterfeiting of machinery`, for which the following two sections have already been drafted:

> *Background Information*: *The Supreme Court confirmed what it had already ruled in previous decisions, ruling definitively on the validity of the XXXX patent and the intervening counterfeiting by YYYY; the plaintiff then filed a petition for the continuation of the suspended trial.*
>
> *Claims and Arguments*: *XXXX seeks relief for the contraction of revenues afferent to the machinery, resulting from the presence of the machinery substitute of YYYY.*

and the *reasoning* section is the active section to be generated. By invoking the Search-by-Content modality for drafting the *reasoning* section, the document builder retrieves as suggestions the most relevant document chunks of the *reasoning* sections of the judgements similar to the active `counterfeiting of machinery` document at hand. The top-2 retrieved document chunks ranked by similarity score are shown below.

> *(...) objected that such conduct challenged to XXXX would not be configurable since the corporate purpose of the two companies is different in the sense that XXXX is limited to carrying out only service activities for the repair of cleaning machinery, while YYYY carries*

> *out as its prominent activity the production and sale of machinery, which XXXX does not deal with, consequently, there could not be talk of unfair competition since the activity carried out by the defendant company is merely marketing and not production pertaining exclusively to the plaintiff company.*
>
> *The plaintiff initially attached only pecuniary damage by referring to the royalties that the defendants would have to pay for the use of the trademark online, then to the cost of advertising investments for the launch of the products then not sold due to unfair competition and alternatively the retroversion of profits. Then referring to the fact that through unfair competition XXXX's turnover would have unlawfully grown by increasing its holding YYYY at the expense of ZZZZ with a reduction for the latter in the number of shares.*

5.3. Search-by-Concept

In the Search-by-Concept scenario, it is possible to ask for text suggestions pertinent to one or more concepts c^* among the ones extracted by ASKE, which are representative of the actual content of the ingested documents. Besides concept(s) of pertinence, the legal actor can also choose the section(s) s^* to which retrieved text suggestions should belong to.

Below, we report an example of retrieved text suggestions pertinent to the concept `public service` located in *background information* section of ingested documents.

> *XXXX sued YYYY and ZZZZ, claiming: that it carries out funeral activities; that, by virtue of Article 5, paragraph 2, R.L. no. 19 of 2004, "in the event that the manager of public cemetery and necropsy services also carries out the funeral activity referred to in Article 13 of this law, corporate separation is mandatory..."; that, in implementation of this legislation, on March 27, 2008 the company YYYY had been established, a company that performs on behalf of the Municipality of Rimini the cemetery and necropsy activity as provided for by Art. 1 point 3 letter c of R.L. n. 19 of 2004 i.e. funeral transport for indigent people, funeral collection and transport on call of the Judicial Authority or for hygienic-sanitary needs, observation depot, morgue, mortuary sanitary service, necroscopic medicine activities; by notarial deed dated 30/09/2009, YYYY conferred the commercial activity concerning funeral services to ZZZZ. (...)*
>
> *The company XXXX reported: that it was dedicated to the provision of logistics and material transport services for the "MotoGP" world championship races and other motorcycle sports; that it had entered into a contract with the organizing company YYYY, thus securing the status of "Express Supplier" and the ability to carry out all transports not only for YYYY, but also for the various teams and suppliers participating in the MotoGP championship; that in the structure of ZZZZ Mr. A. B. has always played a key role, following the conclusion of a project collaboration contract, renewed until 12/31/2008, by which he had been given the position of project manager organizing logistics for the MotoGP and other motorsport world championships, including the contract acquisition stages, with qualifications general director adviser and general director marketing and full decision-making and signing powers in the name and on behalf of ZZZZ; (...)*

6. Related work

Work related to the legal document building issues regards *legal document assembly*, *legal information retrieval*, and *document segmentation*. The increasing interest towards AI applications in the legal field pushed a significant interest towards legal document assembly [2] by relying on a predefined document structure. In [8] the authors define a clear distinction between a document-oriented approach and an issue-oriented approach to document assembly. The former consists in the automatic selection of document components and the instantiation of user provided values for certain variables. The latter makes use of an explicit representation of legal rules, whose truths of the predicates are provided by the judge in the justification. These rules have been explicitly defined over the years in different forms, such as decision trees [9] and interchange languages [4]. Different collection of legal documents, such as Serbian case laws [10] and the GDPR [11], have been modeled using these approaches.

Legal information retrieval (LIR) is the discipline that aims at extracting information from a corpus of legal documents, including case law decisions and legal codes. The digitization of these documents produced a significant boost to the field of LIR, with many methodologies being proposed for the problem: from boolean and rule-based approaches [12], from the exploitation of thesauri [13] to ontologies [14]. Recent studies [15] [16] have been focused on the use of large language models (LLM) [17], that are capable of capturing the contextual representation of the text rather than simply focusing on the occurrences of certain terms. There have also been studies exploiting NLP techniques directly on Italian legal documents: ontology learning systems [18], article prediction [19], and fine-tuned LLMs [20].

The segmentation of legal documents is a task that can be performed either manually or automatically, with the respective advantages and disadvantages. Manual segmentation [21] [22] relies on annotations performed by field experts on a corpus that do not contain many documents due to the complexity of the task, although the results are more precise than any automated system. Automatic approaches to text segmentation have been developed for general corpus in order to detect a shift in the topic discussed by subsequent sentences. In [23] semantic related graphs, in which sentences are nodes and edges represents their relatedness, are exploited in order to find maximal cliques. More recent studies approach the task employing global [24] and contextual [25] vector representations of sentences. Automatic approaches have also been developed for the segmentation of legal documents, such as US Court Decisions [26], and Terms-of-Service documents [27].

7. Concluding remarks

In this paper, we presented a reference service architecture for knowledge-based legal document building. After discussing components and techniques composing the knowledge-extraction service and the suggestion-extraction service of the proposed architecture, some examples of the experimentation of a first legal document builder prototype on a corpus case study in the Italian legal domain are discussed. Ongoing work is devoted to the consolidation of the functionalities of the suggestion-extraction service of the document builder, with development of a graphical interface for the knowledge-based document building functionalities. Future work will be devoted to studying automated document segmentation techniques. In particular, we envision a clause-level classification model, which first maps each clause into a vector space exploiting a pre-trained contextual embedding model (e.g. Sentence-BERT [6]) and then applies a classifier on the clause embedding vectors to determine the segment label of each clause. We plan experimentation of different classifiers, operating either on single clauses, clause sequences or clause networks.

Acknowledgments

This work is partially supported by i) the Next Generation UPP project within the PON programme of the Italian Ministry of Justice, and ii) the project SERICS (PE00000014) under the MUR NRRP funded by the EU - NextGenerationEU.

References

[1] M. Oswald, Algorithm-assisted decision-making in the public sector: framing the issues using administrative law rules governing discretionary power, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 376 (2018) 20170359.

[2] T. Gordon, A theory construction approach to legal document assembly, in: Pre-Proceedings of the Third International Conference on Logic, Informatics, and Law, Citeseer, 1989, pp. 485–498.

[3] G. Pinotti, A. Santosuosso, F. Fazio, 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, pp. 112–121.

[4] M. Palmirani, G. Governatori, A. Rotolo, S. Tabet, H. Boley, A. Paschke, Legalruleml: Xml-based rules and norms., RuleML America 7018 (2011) 298–312.

[5] H. Kozima, Text segmentation based on similarity between words, in: Annual Meeting of the Association for Computational Linguistics, 1993.

[6] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Conference on Empirical Methods in Natural Language Processing, 2019.

[7] V. Bellandi, S. Castano, P. Ceravolo, E. Damiani, A. Ferrara, S. Montanelli, S. Picascia, A. Polimeno, D. Riva, Knowledge-Based Legal Document Retrieval: A Case Study on Italian Civil Court Decisions, in: Proc. of the 1st Int. Knowledge Management for Law Workshop (KM4LAW), Bozen-Bolzano, Italy, 2022.

[8] L. K. Branting, An issue-oriented approach to judicial document assembly, in: Proceedings of the 4th international conference on Artificial intelligence and law, 1993, pp. 228–235.

[9] D. B. Evans, Artificial intelligence and document assembly, Law Prac. Mgmt. 16 (1990) 18.

[10] M. Marković, S. Gostojić, Knowledge-based legal document assembly, arXiv preprint arXiv:2009.06611 (2020).

[11] L. Robaldo, C. Bartolini, G. Lenzini, The dapreco knowledge base: representing the gdpr in legalruleml, in: Proceedings of The 12th Language Resources and Evaluation Conference, 2020, pp. 5688–5697.

[12] W. Y. Mok, J. R. Mok, Legal machine-learning analysis: First steps towards a.i. assisted legal research, in: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law, ICAIL '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 266–267. doi:10.1145/3322640.3326737.

[13] M. Klein, W. Steenbergen, E. Uijttenbroek, A. Lodder, F. Harmelen, Thesaurus-based retrieval of case law., 2006, pp. 61–70.

[14] S. Castano, A. Ferrara, M. Falduti, S. Montanelli, 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, ICAIL '19, ACM, New York, NY, USA, 2019, p. 179–183. doi:10.1145/3322640.3326730.

[15] W. Hu, S. Zhao, Q. Zhao, H. Sun, X. Hu, R. Guo, Y. Li, Y. Cui, L. Ma, BERT_LF: A similar case retrieval method based on legal facts, Wireless Communications and Mobile Computing 2022 (2022) 1–9. URL: https://doi.org/10.1155/2022/2511147. doi:10.1155/2022/2511147.

[16] A. Ferrara, S. Picascia, D. Riva, 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 (JUSMOD22), Hyderabad, India, 2022.

[17] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[18] A. Lenci, S. Montemagni, V. Pirrelli, G. Venturi, Ontology learning from italian legal texts, in: Law, Ontologies and the Semantic Web, IOS Press, 2009, pp. 75–94.

[19] A. Tagarelli, A. Simeri, Unsupervised law article mining based on deep pre-trained language representation models with application to the italian civil code, Artificial Intelligence and Law (2021) 1–57.

[20] D. Licari, G. Comandè, Italian-legal-bert: A pre-trained transformer language model for italian law (2022).

[21] P. Kalamkar, A. Tiwari, A. Agarwal, S. Karn, S. Gupta, V. Raghavan, A. Modi, Corpus for automatic structuring of legal documents, arXiv preprint arXiv:2201.13125 (2022).

[22] G. Zhang, P. Nulty, D. Lillis, A decade of legal argumentation mining: Datasets and approaches, in: Natural Language Processing and Information Systems: 27th International Conference on Applications of Natural Language to Information Systems, NLDB 2022, Valencia, Spain, June 15–17, 2022, Proceedings, Springer, 2022, pp. 240–252.

[23] G. Glavaš, F. Nanni, S. P. Ponzetto, Unsupervised text segmentation using semantic relatedness graphs, in: Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics, Association for Computational Linguistics, 2016, pp. 125–130.

[24] O. Koshorek, A. Cohen, N. Mor, M. Rotman, J. Berant, Text segmentation as a supervised learning task, arXiv preprint arXiv:1803.09337 (2018).

[25] A. Solbiati, K. Heffernan, G. Damaskinos, S. Poddar, S. Modi, J. Cali, Unsupervised topic segmentation of meetings with bert embeddings, arXiv preprint arXiv:2106.12978 (2021).

[26] J. Savelka, K. D. Ashley, Segmenting us court decisions into functional and issue specific parts., in: JURIX, 2018, pp. 111–120.

[27] D. Aumiller, S. Almasian, S. Lackner, M. Gertz, Structural text segmentation of legal documents, in: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, 2021, pp. 2–11.