



An entity-centric approach to manage court judgments based on Natural Language Processing

Valerio Bellandi ^{a,1}, Christian Bernasconi ^{b,1}, Fausto Lodi ^{b,1}, Matteo Palmonari ^{b,1},
Riccardo Pozzi ^{b,1,*}, Marco Ripamonti ^{b,1}, Stefano Siccardi ^{c,1}

^a Computer Science Department, Università degli Studi di Milano, Via Celoria 18, Milan, 20122, MI, Italy

^b Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano Bicocca, Viale Sarca, 336, Milan, 20125, MI, Italy

^c Consorzio Interuniversitario Nazionale per l'Informatica, Via Ariosto, 25, Rome, 00185, RM, Italy

ARTICLE INFO

Keywords:

Legal knowledge extraction
Semantic search
Named Entity Recognition
Zero-shot learning

ABSTRACT

In this paper, we present an entity-centric infrastructure to manage legal documents, especially court judgments, based on the organization of a textual document repository and on the annotation of these documents to serve a variety of downstream tasks. Documents are pre-processed and then iteratively annotated using a set of NLP services that combine complementary approaches based on machine learning and syntactic rules. We present a framework that has been designed to be developed and maintained in a sustainable way, allowing for multiple services and uses of the annotated document repository and considering the scarcity of annotated data as an intrinsic challenge for its development. The design activity is the result of a cooperative project where a scientific team, institutional bodies, and companies appointed to implement the final system are involved in co-design activities. We describe experiments to demonstrate the feasibility of the solution and discuss the main challenges to scaling the system at a national level. In particular, we report the results we obtained in annotating data with different low-resource methods and with solutions designed to combine these approaches in a meaningful way. An essential aspect of the proposed solution is a human-in-the-loop approach to control the output of the annotation algorithms in agreement with the organizational processes in place in Italian courts. Based on these results we advocate for the feasibility of the proposed approach and discuss the challenges that must be addressed to ensure the scalability and robustness of the proposed solution.

1. Introduction

Legal documents, especially court judgments and similar resolutions such as orders, contain information that is valuable in multiple applications (from legal case retrieval to legal process analysis), for different purposes (from the comparison with similar cases for uniform judgments to discovering trends for specific legal subjects, e.g., counting the average maintenance in relation to economical conditions of the partners in divorces), and stakeholders (from judges to law administrators, lawmakers, lawyers, scholars, and the general public). For the sake of simplicity, in the rest of the paper, we will use “judgements” to indicate also similar documents. Several applications, e.g., legal search engines, are already in place, but recent advances in data management, Natural Language Processing (NLP), and Machine Learning (ML) are dramatically transforming legal text processing, promising more powerful or completely novel functionalities in legal applications.

References to entities such as organizations, persons, locations, dates and money² play an important role in these documents: the extraction, consolidation and storage of these references in the form of metadata and annotations can support many of current and future applications. For example, considering entities in a faceted search application [2] can help users retrieve legal cases they are interested in (e.g., “find all judgments that mention the Wells Fargo bank”) or locate specific entities in long judgments (e.g., “find all occurrences of *Jane Smith* in a judgment consisting of several pages”); studying the relation between maintenance and economical conditions of the partners can be better supported by extracting all references to money in the judgment; studying trends at the regional level can be better supported by extracting addresses in multiple judgment. Solutions to extract information from legal documents have in fact a long tradition and are attracting an even increased interest [3].

* Corresponding author.

E-mail address: riccardo.pozzi@unimib.it (R. Pozzi).

¹ All authors contributed equally.

² In this paper we refer to the broad interpretation of “named entities” that is commonly used in the NLP community [1]

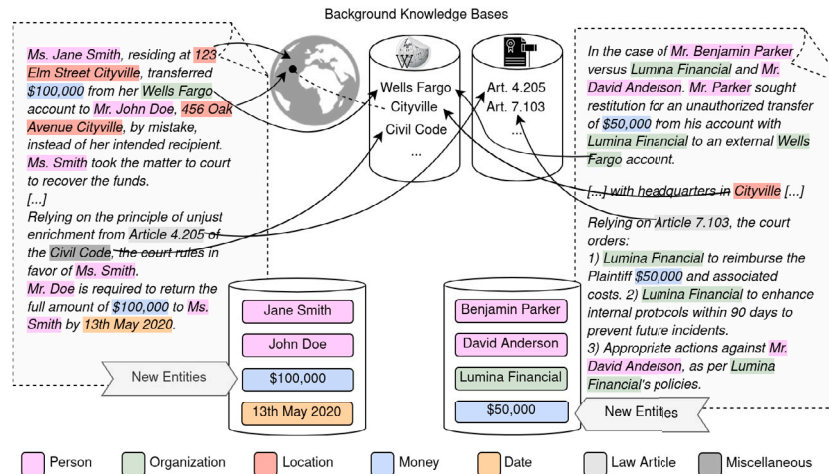


Fig. 1. A sketch of the result of entity extraction and knowledge consolidation. Mentions of entities described in background knowledge bases are linked (links are shown explicitly). Mentions of the same new entities are clustered (links between the mentions and the new entities are persisted but omitted in the figure).

Many approaches have been proposed to extract abstract concepts [4] or named entities [3]. However, most of the existing approaches (see also Section 2.2) addressed only the identification and classification of entity mentions within each document (e.g., the sequence “Ms. Jane Smith” refers to an entity of type “person”), a task known as Named Entity Recognition (NER). However, NER alone falls short of guaranteeing certain functionalities. For instance, while NER can identify and classify entities such as persons or organizations within a text, it does not provide the capability to determine which of these mentions correspond to the same entity. As a consequence, it would not be possible to develop some of the search functionalities discussed before (e.g., find all occurrences of *Jane Smith* in one or multiple judgments). To support similar functionalities, an additional knowledge consolidation process is required. Such a process can be realized by assigning each entity mention an entity identifier, which represents the entity the mention refers to. The application of this entity-centric knowledge consolidation process to legal documents and, in particular to court judgments, has been limited so far.

Also, most of the existing work has focused on algorithms for entity extraction without discussing which kind of software architectures should be developed to implement the extraction processes and manage the data generated thereof.

The main objective of this paper is to investigate the feasibility and implications of supporting entity-centric data enrichment and management for legal documents, focusing on court judgments, and addressing two specific research questions.

1. RQ1 - Can we exploit algorithms that not only find and classify entity mentions but also consolidate the extracted information by grouping entity mentions referring to the same entity, therefore supporting entity-centric data management in downstream applications?
2. RQ2 - Which architectural design can ensure the scalability, extensibility, and availability of entity extraction and consolidation processes in the context of court judgments and downstream applications that rely on annotated documents?

In the paper, we provide a few contributions to answer these research questions. The main idea behind our paper is exemplified in Fig. 1. The figure depicts two fictional judgments and three background KBs: Wikipedia, the geo-coordinate system (geocoding addresses returns points in this system), and a law database; extracted entity mentions are highlighted in different colors depending on their type; after knowledge consolidation, a set of entities appearing in the judgments are identified, some belong or are linked to the background KBs, and some are new entities specific to the processed judgments.

In relation to RQ1, we propose an entity extraction approach that supports entity-centric knowledge consolidation by exploiting background information sources that provide global, canonical, entity identifiers when possible (e.g., geographic coordinates, web identifiers of law articles, Wikipedia webpages), and introducing novel entity identifiers for entities not described in the background sources. While most of the entities, especially persons and organizations, fall in the latter case, canonical links cover 22% of persons, organizations, and locations in the data used in our experiments. To implement this approach, we develop and evaluate a pipeline that uses ML-based and rule-based algorithms to identify and classify entity mentions, predict links to canonical identifiers, and cluster mentions that are not linked and therefore represent “novel” entities. The pipeline is based on approaches proposed in previous work [5–7] but the main focus of this paper is to provide an in-depth analysis of its performance on court judgment data, which, to the best of our knowledge, has not been attempted before. To this end we have developed a gold standard, which we document in the paper; while the gold standard cannot be shared because of data sensitivity reasons, our methodology can be replicated on other similar datasets. The results suggest that the maturity of NLP technologies is quite close to supporting full-fledged entity extraction and consolidation for similar legal data. However, a few challenges – discussed in the paper – must be addressed to deliver solutions that can be really adopted in courts and trials at the national level.

In relation to RQ2, we propose an architecture that is intended to capture domain requirements collected in the context of two projects developed in cooperation with or funded by DGSIA³ and the Ministry of Justice. In particular, our proposal is aimed at supporting entity extraction and consolidation, explicitly considering: access to the different information pieces that compose the enriched data (e.g., search), the revision of the output of the algorithms in a human-in-the-loop fashion, the addition and modification of processing components in a modular fashion (extensibility), and the scalability of algorithm execution.

The paper is structured as follows: we begin by discussing related work in Section 2 and the data used in our work in Section 3. In Section 4, we describe the pipeline proposed for addressing RQ1, while Section 5 covers the architecture proposed for addressing RQ2. Our experiments are detailed in Section 6. Finally, we discuss conclusions and future work in Section 7.

³ The institutional body that manages the information systems of the Ministry of Justice.

2. Related work

We organize the discussion about related work in three sections. In Section 2.1, we introduce the conceptualization of entity extraction and knowledge consolidation approaches into a set of different tasks and discuss the latest advancements without specific reference to the legal domain. In Section 2.2, we discuss the application of entity extraction and knowledge consolidation approaches in the legal domain. In Section 2.3, we discuss architectures proposed to support these approaches in the legal domain.

2.1. Entity extraction and knowledge consolidation: Tasks and approaches

In the literature, the extraction of entity mentions from text and the subsequent knowledge consolidation process are often conceptually divided into different tasks [8].

The *Named Entity Recognition (NER)* task consists of the identification of the sequences of tokens that correspond to entity mentions (spans) and the classification of these mentions into a predefined set of classes (e.g., “person”, “location”, and “organization”); most recent approaches treat the task as a supervised sequential classification problem, often leveraging pre-trained large language models as feature extractors.

Several approaches consider also the problem of consolidating the extracted entity mentions to identify which ones refer to the same entities, which we refer to as the knowledge consolidation problem. Some approaches propose to consolidate knowledge by clustering the entities found [9] without exploiting any background Knowledge Base (KB), while most of the approaches exploit background KBs [5–7] to improve data integration by interlinking documents referring to entities described in these KBs. The latter approaches have also the advantage that they make it possible to exploit the vast amount of knowledge stored in KBs to further enrich the data (for example, entities in Wikipedia are classified into categories). In our work, we choose this second KB-supported approach and use the conceptual framework proposed in Knowledge Base Population track of the Text Analysis Conference’s (TAC-KBP)⁴ [8]. In this framework, three sub-tasks are considered: Named Entity Linking, NIL Prediction, and NIL Clustering. Despite the terminology may slightly change, these tasks are implemented in similar approaches to knowledge consolidation [5–7].

Named Entity Linking (NEL) consists of linking entity mentions to their corresponding entities in the KB, e.g., the mention “Wells Fargo” corresponds to the Wikipedia entity described in the page https://en.wikipedia.org/wiki/Wells_Fargo (all the examples here are referred to Fig. 1). It is possible, however, that an entity mention refers to an entity that is not present in the KB, in which case, it should be tagged as NIL (“Not in Lexicon”). This task consisting of predicting whether an entity is NIL, sometimes interpreted as sub-task of NEL, is frequently referred to as *NIL Prediction*; for example, NIL prediction should tell for all mentions which should be linked, e.g., each mention of “Cityville”, and which should not and tagged as NIL, e.g., each mention of “Jane Smith”. While in KB-supported consolidation all mentions linked to some KB entity (e.g., the mentions of “Cityville”) are implicitly clustered, NIL entity mentions (e.g., the mentions of “Jane Smith”) are not; therefore a final task consists of *NIL Clustering*, i.e., clustering all the NIL entity mentions that refer to the same entity. In the example shown in Fig. 1, this KB-supported knowledge consolidation process is expected to extract mentions of five entities described in two different KBs (Wikipedia and a database of Law Articles) and mentions of eight new entities. It is worth observing that while NIL Prediction and NIL Clustering received less attention than NEL until a while ago (e.g., see [10]), the interest in these tasks has significantly increased in recent work [6,7], because of their centrality in entity

extraction frameworks. In the following paragraphs, we briefly describe recent developments related to each of these tasks regardless of their application to the legal domain and summarize the relation between these developments and the approaches used in our paper.

NER. Several approaches based on rules or machine learning with manually engineered features have been developed and are still deemed to be helpful in practice for some types of entities, which are named using a set of known patterns [11]. However, latest NER approaches exploit deep learning methods possibly combined with conditional random fields (CRFs) [11]. Word and character embeddings provide valuable features to support a NER classifier, which performs a sequence tagging task. The leading approach on the CoNLL2003 NER benchmark for the English language obtains an F_1 score of 94.6 using a mix of character-based, contextualized, and non-contextualized embeddings [12].

However, recent work showed BERT models [13] outperform CRFs on Italian NER datasets considering the types “person”, “organization”, and “location” [14]. In our work, we test the combination of existing NER models for the Italian language based on rules, CRF, and contextualized word embeddings, pre-trained and fine-tuned.

NEL. In the last decade, techniques for NEL began using representation learning to obtain dense representations of mentions (in their context) and entity descriptions and compute their similarity scores [15,16]. Attention mechanisms and transformers [17] have improved techniques based on dense representations, leading to the emergence of the bi-encoder and cross-encoder paradigms for dense-retrieval and candidate re-ranking, respectively. These paradigms have been deeply explored in BLINK [10] and extensively used afterwards [5–7]. Lately, some novel NEL paradigms are emerging: autoregressive entity linking [18] and extractive entity linking [19]. In our work, we use a bi-encoder based on BLINK, trained on the Italian Wikipedia, which, as we argue later on in the paper (see Sections 4 and 6) provides a good trade-off between performance and scalability [7].

NIL prediction. NIL prediction approaches, many originating from TAC-KBP, include techniques such as setting a score threshold on the NEL score, incorporating an extra class for NIL, and utilizing a binary classifier with the linking score and additional features as input [11]. More recently, the task has been addressed in two kinds of approaches based on BLINK and evaluated on English benchmarks: the first kind considers BLINK scores as features to classify mentions as NIL [5,6] (see Section 4 for more details), while the second one clusters the mentions’ representations to detect NIL entities [7]. A performance-wise comparison of the two approaches is not available. In our work, we use the first kind of approach because it can be executed on individual mentions avoiding clustering.

NIL clustering. Several NIL Clustering approaches derive from TAC-KBP editions, including methods based on lexical similarity [20], character or word-based embeddings [21], or the combination of hand-crafted features and word-embeddings [22]. NIL clustering is similar to *coreference resolution*, a task consisting of identifying mentions (NIL and non-NIL) that refer to the same entity. Indeed, most recent NIL clustering approaches [6,9] derive from this thread. In our work, we use a method specific for NIL Clustering that combines BLINK-based dense representations (as used in the latter approaches) and lexical features (as used in the first approaches) [5].

2.2. NLP, entity extraction and knowledge consolidation in the legal domain

Several approaches based on NLP have been proposed to improve the management of legal documents in different use cases [23], including legal case summarization [24] and retrieval [25,26], legal rule extraction [27], and data management for criminal investigations [28–30]. The approaches proposed for criminal investigations are based on entity-centric data management principles; therefore, they apply

⁴ <https://tac.nist.gov/>

entity extraction techniques and, in some cases, describe downstream applications where the extracted entities are used to support faceted search engines [28]. However, these approaches apply entity extraction to sources that are relevant for the investigations (e.g., scraped web pages, images, police reports) but not proper legal documents like court judgments, which represent specific legal language distributions. Also, one of these approaches [28] does not apply end-to-end entity extraction pipelines for generic entities as we do in our work, but focuses on specific pattern-based extraction rules. The approach proposed in [30] admittedly discusses preliminary experiments and baseline algorithms. Our work takes inspiration from these approaches, especially in terms of data management principles and background architecture, but we target different types of data, especially court judgments; with respect to [30], we also use novel algorithms and a more solid evaluation.

Several approaches have focused on the application of entity extraction to legal data similar to ours, and, specifically, court judgments. Most of these approaches have applied NER techniques, which support a variety of downstream applications including anonymization, an objective of interest for most legal systems worldwide. A thorough review of previous work on NER applied in the legal domain can be found in [3]. In order to avoid manual labeling, an approach developed for Indian Supreme Court Judgments adopts a rule-based approach to information extraction [31]; it classifies entities into domain-specific types and extracts relations based on a custom ontology. The approach has been evaluated on five judgments, obtaining high precision but very limited recall for most of the entity types, a known bottleneck of rule-based methods. However, combining different algorithms, including rule-based ones, was found to be beneficial for NER in the legal domain [32,33]. In our paper, we evaluate several NER algorithms that can be considered competitive on generic texts on Italian court judgments, including algorithms based on contextualized word embeddings fine-tuned on our gold standard. Also, considering the evidence reported in previous work, e.g., in [31], we prefer to focus now on assessing and improving the extraction of a limited number of rather generic entity types (see Section 3.1); we capture some specific types of entities with rules (as discussed in Section 6) and we leave the task of finer-grained entity classification for future work, a task for which promising zero-shot methods are emerging [34].

A few approaches consider NEL in legal texts. An approach applied NER and NEL on a corpus of judgments of the European Court of Human Rights [35], using a legal ontology as background KB. Another approach proposes to apply NEL on the EUR-Lex law article dataset [36]; it is trained using transfer learning. In our work, we study an end-to-end combination of NER, NEL, NIL prediction, and NIL Clustering for full-fledged knowledge consolidation and we use a more recent pre-trained NEL approach [10]. Another study combined BERT with rule-based techniques for NER and coupled it with an off-the-shelves NEL service to extract entities from court decisions in the Finnish language [37]. This study is more similar to ours because it also applies, to some extent, a knowledge consolidation approach; however, we also focus on NIL Prediction and NIL Clustering, and we use a BLINK-based NEL algorithm. Some work combining NEL and NIL prediction has been evaluated on historic legal documents [38] (the depositions of the 1641 Irish rebellion⁵). However, to the best of our knowledge, no prior work investigated the combination of NER, NEL, and NIL Prediction in recent court judgments.

Some approaches have focused on the extraction of abstract concepts and legal terminology [4], or other kinds of information. For example, a knowledge management system is proposed to semi-automate the extraction of norms and their elements and populate legal ontologies [39], based on Semantic Role Labeling. Their approach consists of general-purpose NLP modules with pre- and post-processing using rules based on domain knowledge. These approaches have orthogonal

objectives compared to ours, which focus on named entities, and could be in principle combined in the future.

Finally, NLP methods have been applied to support legal information retrieval and develop search engines. A few approaches focus on leveraging text summarization techniques [24,26,40], but information extraction and retrieval are interconnected tasks [4], as appears from contributions to the Competition on Legal Information Extraction/Entailment (COLIEE) organized since 2017 [41]. In this paper, we do not focus on specific algorithms for legal information retrieval; however, our entity extraction approach and architecture are intended to support document search by exploiting entity-related annotations, e.g., to populate search facets and filter out documents.

2.3. Architectures and entity-centric infrastructures for the legal domain

The body of work discussed in the previous sections has paid more attention to conceptual frameworks, algorithms, and their evaluation, and less attention to architectures to support the extraction processes and to manage their output. However, a few studies have addressed these problems from an architectural point of view in recent years. In [42], the authors conducted a systematic mapping study to identify and analyze state-of-the-art software architecture for NLP, in the field of legal documents. They analyzed relevant papers and identified several architectural approaches, including architectures based on pipelines, services, and microservices. They have found that the last approaches focus especially on pipelines, which involve a sequence of NLP modules that process text in a predefined order. The authors also argue that service-oriented and microservices architectures have advantages in terms of flexibility and scalability. However, they do not identify a generic infrastructure to manage entities in the legal domain as we do in our work (see Section 5).

In the Ref. [43], the authors present a software architecture designed to aid legal professionals in resolving legal cases through automated extraction of crucial information from documents and generation of potential arguments. This system employs a fusion of rule-based and statistical natural language processing techniques, with a specific focus on a dedicated knowledge extraction pipeline.

Certain architectures seamlessly integrate NLP services and ontologies. For example, in [44], a document management system within the legal domain is elaborated, demonstrating the conversion of diverse paper documents into RDF statements. This transformation enables efficient indexing, retrieval, and long-term preservation. Another architecture, discussed in [45], concentrates on the analysis and extraction of specific entities from legal texts like acts and agreements. This approach relies on an ontology containing pertinent information about document types, their structures, the entities to be extracted from each section, and the requisite processing steps involving a pre-existing text analysis library. The enhancement of system performance led to the adoption of a microservices-based architecture integrated with message brokers, as detailed in [46]. This implementation was integrated into a high-level document management system used for extensive language analysis of legal documents. It is worth noting that although this implementation shares some characteristics with our own design, it does not encompass the knowledge consolidation process central to our work.

3. Data collection and gold standard

The creation of a gold standard is an essential part of the process required to adapt entity extraction and knowledge consolidation algorithms to the legal domain, assess their performance, and drive their improvement. In this section, we discuss the methodology used to create this gold standard from a larger corpus and its features.

We collected a corpus of 927,453 real judgments in civil trials published from 2008 to 2021 (the majority of the judgments, ~86%, are published from 2016 to 2021). This number can be compared to

⁵ <http://1641.tcd.ie/>

the total number of trials that are estimated per year according to [47], which falls between 2 and 2.5 millions from 2010 to 2019.

Data includes the judgment text and 41 metadata with information about the judge (or the president if several judges are involved), the number and year of the judgment and of the trial, the court and the district it belongs to, the instance (trial or appeal), references to the trial in case of appeal, a code describing the subject and some technical fields of no interest here. This dataset has been provided by the Ministry of Justice and consists of real documents as they were archived. As a consequence, in many cases, the texts contain spurious lines that must be cleared before processing, for instance, some metadata at the very beginning, duplicate headings, extra blank lines or characters from stamps present in the printed version.

The judgments' structure consists of several sections, the most important being: (i) a preamble with the judge(s), plaintiff(s) and defendant(s) data, (ii) the description of the case, (iii) the final decision and dispositions with the related reasons.

3.1. Gold standard

From the corpus of 927,453 court judgments described above, we defined a gold standard consisting of 146 labeled documents selected using stratified sampling on the province of the court that made the judgment. The annotation was performed semi-automatically: (i) the documents were annotated with extraction algorithms, then (ii) the human annotators reviewed the automatic annotations. This approach, where algorithms are used to speed up the annotation process, has been used for developing a NER gold standard in previous work [14].

The annotations include labels for NER, NEL, NIL prediction, and NIL clustering.

These tasks are executed sequentially, thus, each step has been performed by manually refining the output of the algorithm of interest applied to the annotated documents from the previous step. For this reason, we can consider the annotation process as a simulation of the real human-in-the-loop use case, thus making it possible to estimate the amount of time required by the final user to consolidate the main pipeline algorithms output.

The annotation process involved the following entity classes:

1. "person",
2. "location",
3. "organization",
4. "money",
5. "date",
6. "miscellaneous".

To obtain an objective measure of the quality of the NER annotations (GS_{NER}), which served as the starting point for subsequent annotation processes, we calculated an inter-annotator agreement (IAA) measure. We then used this gold standard as a basis for creating GS_{CON} , which includes annotations for the remaining tasks, also referred to as the consolidation process.

3.1.1. NER

Annotation process. The annotation process for GS_{NER} has been carried out by two annotators using Doccano⁶, a web application for labeling tasks. Each annotator was asked to annotate 88 documents, of which 15 were overlapping with the other annotator so that 30 documents (15+15) contained annotation from both. These overlapping documents are necessary for calculating the inter-annotator agreement (IAA). For this specific task, an annotation is composed of the index of the start character, the end index, and the type of the mentioned entity.

Based on [48], a detailed set of guidelines has been prepared and provided to the annotators to reduce as much as possible the inconsistencies that could be generated by different styles of annotation. During the annotation process, we iteratively refined the guidelines by keeping track of the doubts cast by the annotators. The guidelines involve both the span selection (e.g., in case of nested entities, annotate the span of the most informative supported type: "Via Garibaldi 18, Roma[LOC]" must be preferred over "Via Garibaldi 18[LOC], Roma[LOC]") and the type assignment (e.g., a type must be assigned based on the context in which the entity appear: "Mario Rossi, born in Italy[LOC]" vs "Italy[ORG] asked the European Union to ..."). Furthermore, each type has its own guidelines to deal with particular cases (e.g., exclude the title of a person from the annotation: "Dr. Mario Rossi[PER]" must be preferred over "Dr. Mario Rossi[PER]"). Furthermore, we decided to annotate mentions such as "the Judge" or "the Court" that are not exactly in the scope of NER but close to the coreference resolution task, because these mentions are relevant to the domain.

Document statistics. The resulting corpus counts 16,634 annotations (~114 annotations per document), with an average document length of ~1900 words. Each document required on average ~15 min to be annotated. The distribution of types is reported in Fig. 2.

Inter-annotator agreement. To assess the quality of the gold standard and the clarity of the guidelines we compute the inter-annotator agreement (IAA) on the 30 overlapping documents, which are annotated independently by both the annotators. Since there is no standard way to evaluate the agreement level of a NER dataset [49], we adopted an IAA based on F_1 score computed with the criteria described in Section 6.1.1. In this context, the F_1 score is pair-wisely computed between annotators: for each combination, an annotator is used as gold standard to evaluate the other annotator; an average is computed in case of more than two annotators.

Popular IAA metrics such as Cohen's Kappa, Scott's Pi, Fleiss' Kappa, and Krippendorff's Alpha, may not be suitable for complex labeling tasks such as NER, providing low interpretability values [49]. As discussed in Section 6.1.1, a problem of this task is that we deal with text spans, so it is difficult to define a general criterion to identify positive and negative examples to calculate a consistent IAA value [50]. A simple solution to face this problem is to compute the metrics at token-level, however this would yield overly optimistic IAA value (e.g., the pair of "Barack Obama[PER] was born in Honolulu" and "Barack[PER] Obama[PER] was born in Honolulu" is considered a perfect match). For these reasons, the choice of an F_1 score-based IAA as the main metric is a more robust alternative to the classic metrics [51,52].

Table 1 shows all metrics averaged over documents. For completeness, we also report the token-level F_1 score for the *strong typed* match criteria and Cohen's Kappa, Scott's Pi, Fleiss' Kappa, and Krippendorff's Alpha. We can see that *strong-typed (instance-level)*, i.e., the strictest metric, has a score of .808, which can be considered satisfactory since it measures perfect matches between annotators. Moreover, by looking at the additional F_1 score-based metrics we can see that the values increase as we relax the matching criterion (partial is the least stringent). This proves that even in cases where the labels from the two annotators do not perfectly match, there is still a significant degree of overlap. These differences between hard and soft constrained F_1 scores, as well as the difference between *instance-level* and *token-level strong typed*, also point out that a perfect span selection was the most difficult part for the annotators, as expected.

3.1.2. Knowledge consolidation

The gold standard GS_{CON} for the knowledge consolidation process, with annotations for NER, NEL, NIL prediction, and NIL clustering, has been created semi-automatically from the 30 overlapping documents used to compute the IAA. The set of general guidelines defined for this task is much simpler than the NER ones. Based on the assumptions in [48], the guidelines can be summed up in two steps:

⁶ <https://doccano.github.io/doccano/>

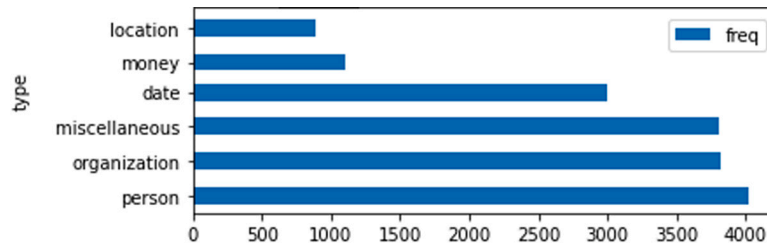


Fig. 2. Distribution of entity mention types in the GS_{NER} gold standard.

Table 1

Inter-annotator agreement computed on a subset of 30 documents from the gold standard GS_{NER}. The description of each F₁ score-based metric can be found in Section 6.1.1. *based on strong typed match.

Metric	Value
F ₁ score-based metrics	
strong	.839
strong typed (instance-level)	.808
strong typed (token-level)	.886
approximate	.966
approximate typed	.904
partial	.969
partial typed	.906
Other metrics	
Cohen*/Fleiss*/Krippendorff*/Scott*	.662

1. if the mention refers to a known entity in the KB (Italian Wikipedia and manually added entities) label it with the entity URI and mark the mention as ~NIL;
2. otherwise, mark the mention as NIL, create a new entity, and label the mention with the new entity URI. This latter step is necessary to trace which NIL mentions refer to the same unknown entity (NIL clustering).

To enhance the efficiency of the annotation procedure a UI was developed, incorporating both Wikipedia search API and a fuzzy search mechanism. On average documents required ~15 min to be annotated for NEL, NIL prediction, and NIL clustering. Out of the 3006 annotated mentions, 467 refer to Wikipedia entities, and 1753 to NIL entities. The remaining 786 mentions, categorized as either *Date* or *Money*, were left unassociated with any entity.

The 2200 mentions linked to an entity are organized into a total of 1025 clusters, with 211 clusters corresponding to Wikipedia entities. The larger portion of 814 refers to NIL entities, as expected within the considered domain.

4. Entity extraction and knowledge consolidation

To address RQ1, inspired by [5–7] we develop a pipeline system that orchestrates first the entity extraction process, i.e., the NER task, and then the knowledge consolidation processes, namely NEL, NIL prediction, and NIL clustering. With NER we use different algorithms to extract all the types annotated in GS_{NER}; the algorithms are described in detail in the remainder of this section.

Table 2 illustrates the supported types for each NER extractor, as well as the NEL, NIL prediction, and NIL clustering algorithms. Notably, for the latter three tasks, we exclusively consider the types “person”, “location”, and “organization”. This decision is based on the fact that “date” and “money” are not directly associated with any linkable entity. With respect to the “miscellaneous” class, during an exploratory phase, we found that the quality of the prediction for this class, significantly lower than for the other classes, was not promising enough to proceed with the knowledge consolidation process. We further discuss this issue in the experimental evaluation.

By using multiple NER algorithms, it becomes necessary to combine the extracted annotations and resolve any conflicts that may arise. To this end, we present heuristic-based combination rules, which are described in the remainder of this section.

4.1. Entity extraction

For the NER task, two general-domain ML-based algorithms for the Italian language have been selected: SpaCy and Tint. In addition, we developed a rule-based service, namely TrieNER, supporting efficient retrieval on a custom dictionary and on-the-fly update of the reference dictionary.⁷ Each algorithm recognizes a different set of types; Table 2 shows which entity types of our gold standard are recognized by each algorithm including the combination of them.

SpaCy⁸ is a NLP library designed to be production-ready, able to perform NER in several languages including Italian. NER is performed by first obtaining a vector representation of the input, in our case with a pre-trained Italian transformer (dbmdz/bert-base-italian-uncased⁹), then the vectors are processed by a neural transition-based parser. We trained the SpaCy NER system on the Italian WikiNER dataset [53], which is composed of annotated Wikipedia articles.

Tint [54] is an Italian NLP library based on Stanford CoreNLP [55]. The NER task is based on Conditional Random Fields (CRF) sequence taggers, combined with rule-based systems to extract money, numbers, and temporal expressions.

TrieNER¹⁰ is a pattern-matching approach, based on a prefix-tree (or trie), that uses a dictionary of entity titles derived from documents metadata. Entities are divided into tokens, personal names are enriched by their permutations, and then the tokens are added to the trie index. This algorithm also identifies partial matches, e.g. the name of a person without the surname. This service is not limited to NER, indeed, besides knowing the matched pattern is part of a named entity we are also aware of the entity (or the entities if both can be mentioned with the same pattern) from which the pattern has been derived. This pattern-based algorithm is used to exploit metadata attached to the judgments, which cover plaintiffs, defendants, and judges. Although these metadata are incomplete, TrieNER is introduced to increase the likelihood that, when an entity appears in the metadata, it is also identified by the NER component. Supposing that “Jane Smith” is listed as a plaintiff in the metadata, TrieNER would search for all the tokens and their permutation in the text, finding mentions like “Ms. Smith” (with a partial match on the token “Smith”) and “Smith Jane”.

Finally, we need to combine the annotations produced by the different algorithms. Each NER annotation consists of a text span associated with a type, thus conflicts may arise when the spans of two annotations overlap (e.g., in “Mr. Cityville” we may have a first algorithm that identifies “[Mr. Cityville (PER)]” while a second algorithm identifies

⁷ On-the-fly dictionary update is not yet active in the current pipeline but we consider it important for future development.

⁸ <https://spacy.io/>

⁹ <https://huggingface.co/dbmdz/bert-base-italian-uncased>

¹⁰ TrieNER is based on trie-search <https://github.com/s-yata/marisa-trie>.

Table 2

Types supported by the NER, NEL, NIL prediction, and NIL Clustering algorithms. *Note that the TrieNER relies on a knowledge base and it is potentially capable of extracting any entity regardless of its type as long as a pattern is provided. In our experiments, TrieNER extracts entities of type “person” and “organization”, since the knowledge base (derived from documents metadata) of reference is limited to these two types.

Types	NER				NEL	NIL pred.	NIL clust.
	SpaCy	Tint	TrieNER*	Combination			
Person	✓	✓	✓*	✓	✓	✓	✓
Location	✓	✓	–	✓	✓	✓	✓
Organization	✓	✓	✓*	✓	✓	✓	✓
Money	–	✓	–	✓	–	–	–
Date	–	✓	–	✓	–	–	–
Miscellaneous	✓	–	–	✓	–	–	–

“[Cityville (LOC)]”) or when the types of two overlapping annotations are inconsistent (e.g., “Italy” identified both as *Location* and *Organization*).

First, we identify span conflicts, then we address type-related conflicts:

1. two overlapping annotations may be totally overlapping (i.e., they share the same exact boundaries) or partially overlapping. In the latter case, we prefer the longest annotation, assuming it is the most informative one. Exceptionally, for the annotations of type “person”, we down-prioritize annotations whose span is longer than k tokens,¹¹
2. with respect to type-related conflicts, different algorithms may predict different types. Consequently, we assign a weight to each algorithm and, in case of type-related conflicts, we perform a *weighted majority vote* to select the type with the highest score (in case two or more algorithms predicted the same type, the score of the type is given by the sum of the algorithms’ weights). This mechanism allows us to control the impact of each algorithm on the final prediction.

4.2. Knowledge consolidation

The knowledge consolidation process involves the three tasks of NEL, NIL prediction, and NIL clustering. NEL is executed by a bi-encoder based on BLINK [10]. The bi-encoder is a method for entity retrieval trained to encode mentions and entity descriptions in the same dense space, in such a way that the distance between a mention and the corresponding entity description is minimized. The training objective is to maximize the similarity of the correct mention-entity pairs; in this case the similarity is calculated with the dot-product and basically represents a linking score.

We chose the bi-encoder paradigm because it is able to produce a semantic representation of both mentions and entities. This property allows us to calculate the semantic similarity of entity mentions during the NIL clustering step and to represent new entities using their mentions so that the NEL system is able to link to the new entities [5].

The training for the Italian language has been performed following the approach of BLINK authors for the English language. We created an entity linking dataset using the hyperlinks from Italian Wikipedia articles as training samples where the anchor text is the mention. Differently from [10], we used $BERT_{BASE}$ ¹² instead of $BERT_{LARGE}$ for computational reasons. The training process has been done in 4 epochs with 9M training examples using in-batch random negatives followed by one epoch of hard negatives in which we use a single hard negative for each training sample. Given a mention, the hard negative is the negative entity (not the correct one) with the highest linking score, calculated using the bi-encoder trained with random negatives.

NIL prediction is implemented as a binary classification using logistic regression. It has been trained with the same Wikipedia-based

dataset used to train the NEL model. Different combinations of input features have been evaluated. The ones we are currently using are the NEL score of the top-ranked entity and the difference between the top score and the score of the second-best candidate.

The NIL clustering sub-service is inspired by a three-step algorithm [56]. First, mentions are clustered based on their surface form, with a tolerance of edit-distance ≤ 3 for words longer than 3 characters, or exact matching for shorter words. Next, a hierarchical clustering algorithm is applied to each cluster, using a predetermined threshold to split clusters based on the semantic representations of the mentions generated by the NEL bi-encoder. NIL clustering thresholds are obtained with grid search using the same Wikipedia-based dataset used to train the NEL model. This step creates sub-clusters within each first-step cluster. Finally, each sub-cluster is represented using the medoid vector of all the mention representations of the sub-clusters, and those that are semantically similar, according to the similarity (dot-product) between the medoid vector, are merged.

5. Architecture

The software architecture is the foundational structure that shapes the design and development of a software system. It serves as the blueprint for creating robust, scalable, and maintainable applications. The main characteristics of software architecture are pivotal in determining how well a system will perform, adapt, and evolve over time. These characteristics define the essence of the architecture and guide decisions throughout the development lifecycle. The main objective of our proposal is to define a full stack system capable of supporting the main functional characteristics such as: (i) document analysis and entity extraction, (ii) centralized annotation management (iii) storage persistent and univocal of the identified entities and (iv) the intra-documental and inter-documental search and entity management functions. Furthermore, the modeling and design of the architecture must also take into consideration characteristics not related to the functionalities, in particular in our case we propose an architecture that takes into consideration the following Non-Functional Requirements (NFR):

- scalability: it must be possible to add documents just increasing storage and computing resources, with no needs of system reconfiguration,
- extensibility: it must be possible to add new document types with their metadata, new User Interfaces and new services, with no impacts on the non involved systems and data,
- availability: subsystems must not suffer from changes, fixes, additions or just operation of other systems.

The main components of our architecture are described in the following paragraphs.

The Document Storage (DS). The DS maintains document texts and metadata as they can be found in the source systems. It is equipped with APIs to import, export, update and query the document using both text and metadata search. Multiple copies of the same document can be stored, e.g. the raw text and preprocessed versions where some data

¹¹ We heuristically found that $k = 6$ fit the court judgment dataset.

¹² <https://huggingface.co/dbmdz/bert-base-italian-uncased>

cleaning has been done. It has been implemented as an ElasticSearch¹³ repository, with the possibility to manage several indexes each with proper metadata to store different document types. All the metadata described in Section 3 have been used for indexing; texts have been cleaned before import.

The Query Layer (QL). The QL interfaces the DS and the annotation database (see below) exposing APIs to query both data, so that client programs do not interface directly the data storage components. The output is in the format used by GateNLP¹⁴ to represent documents, annotations and document features. Presently the QL is not equipped with a complete Access Control mechanism or component, just a simple login mechanism is provided. The main API is used to query the documents choosing logical expressions for the metadata, annotations and text content, with the option to choose which metadata must be returned; pagination is supported. Another API is used to add annotations.

The Front End User Interfaces (UI). Several UI have been provided. The **Search UI** prompts the user to enter the metadata and annotations to query, with values and logical operators, the word or sentence to look for in the document text and the data to display. The output is shown in tabular form, with links to show the raw text and to open the UI for single documents. Then the user may select to run an analysis (performed by a Service System, see below) on the query result. The UI interfaces the Request Management to queue the request. A dedicated management section of the UI shows the pending, running and terminated requests with messages and results details.

The Document UI receives the annotated document GateNLP data from the QL and displays the text highlighting the annotations, that are quotes of entities. The user can choose the annotation sets to display/hide (i.e., showing annotations obtained using different algorithms); clicking an annotation shows details, such as its type hierarchy, links to external sources (e.g. Wikipedia, Google Maps, the database of Italian laws, etc.); alternative links. Users can modify or delete annotations. They can create new annotations and new annotation types. Moreover the UI shows the clusters of entities quoted in the document. Entities, grouped by type, are shown in a side bar and can be expanded to show clusters (e.g. the cluster of mentions of John Smith) and to navigate to the related text section.

The Data Scientist UI is just a general purpose notebook, that needs programmer skills by the user. The **Entity Registry UI** has both functions to query and maintain the ER metamodel, and functions to manage the entities, that is to search entities in the documents, highlight their attributes, perform merge and split operations.

The Request Management (RM). It is used to request that a specific service is run against a set of documents. Its APIs are called by the UI when the user wants to create a request or to check the status of their previous requests. It must be able to store parameters provided by the users and to forward them to the service systems (SS). This may happen using queues or any other suitable mechanisms. It has been implemented as a queue system, using Apache Kafka, an open-source distributed event streaming platform, where the SS can find parameters to get the documents to work. It maintains a table with tasks, their results and statuses.

The Annotation Database (AD). It stores annotations computed by the service systems. Annotations refer to specific portions of document texts and hold information about entities, sentences, sections and so on. The AD has been implemented as a SQL database. The main columns are: the identifiers of the SS that created the annotation and of the document where it was found; the start and end position in the document; the annotation keyword (e.g., “person”, “organization”,

etc.) and value; a field for extra features, in json format, that depend on the annotation type (e.g. external links, notes, etc.)

The Service Systems (SS). Any SS receive as input texts and annotations and compute either new annotations or new text versions (e.g. after cleaning some type of garbage or summarizing and so on). They use the QL APIs to get the data and to store their output, read parameters from the RM system and expose standard APIs to be called by it.

Examples of Service Systems include algorithms that extract fixed format expressions, such as Italian Fiscal Codes, car plates, and phone numbers, but also algorithms that extract more complex and variable expressions like postal addresses and references to law articles.

Finally, a basic rule-based cleaning service was provided, that removes unnecessary line feeds, page headings (e.g. page numbering, and so on), and repeated sets of lines of a few characters. This last case applies when a page, that has been scanned and processed by an Optical Character Recognizer, contained a vertical writing like a long stamp in the margin.

The Entity Registry (ER). The ER is a system where each entity found by the SS and stored in the AD has a unique entry used for disambiguation. Moreover, queries based on ER entries can be run against all the documents in the system. A general ER model has been described for instance in [58] and an application to another legal context in [30]. It consists of a metamodel and a database of entries. The metamodel stores the entity types and, for each type, the set of attributes that suffice to completely identify an instance. An example of an entity type is “person” whose identifying set of attributes might be *personal code* and the tuple (*first name(s)*, *last name*, *birth date*, *birth place*). The ER exposes APIs to store, retrieve and manage both the metamodel and the instances. The ER has been implemented as a graph database using Neo4j¹⁵ as database manager. A dedicated server interfaces Neo4j and exposes both APIs to manage the metamodel and APIs to manage the entities. When an entity is created, the ER assigns a unique identifier to it. Some special APIs are provided, for instance to **merge** two entities that have been belatedly recognized to be the same in the real world or to **split** one, when two real world entities have been erroneously considered the same.

The Service Catalog (SC). The SC stores addresses and functions of the available services and of the SS that provide them. The kernel of the SC is a table holding the addresses of the SS, with the names and parameters of the services they provide. When a new SS is added to the catalog, it is immediately callable by the RM service.

The Service Orchestrator (SO). It manages workflows of SS to be applied at document sets and to schedule periodic operations, or task needed when a triggering event happens. Using the SO, it is possible to combine any services to implement complex workflows; for instance we created a workflow to feed the ER with entities acting as plaintiff, defendant or lawyers. The first service finds the preamble in the judgment; then are run services to find persons and organizations, fiscal codes, dates, places, and postal addresses in the preambles. Finally another service links each person with their fiscal codes or place and date of birth; each organization with its fiscal code; each person and organization with their address (if specified). Resulting persons and organizations are added to the ER, using their names and fiscal codes or birth data as identifiers.

Final Considerations. The situation is sketched in Fig. 3, that highlights the main flows of data and control. The central role of the UI reflects the importance of the user in the whole process of knowledge extraction, especially for the validation and management of annotations (so called *human-in-the-loop*).

The requirement of **scalability** is addressed by the implementation of the Document Storage and the Annotation Database. The requirement of **extensibility** to new types of documents and metadata is

¹³ ElasticSearch (<https://www.elastic.co/>) is a distributed, free and open search and analytics engine for textual, numerical, geospatial, structured, and unstructured data.

¹⁴ where GATE stands for General Architecture for Text Engineering, a suite of tools for natural language processing; see e.g. [57]

¹⁵ A graph databases stores nodes and relationships instead of tables or documents and Neo4j (<https://neo4j.com/>) is a well known implementation.

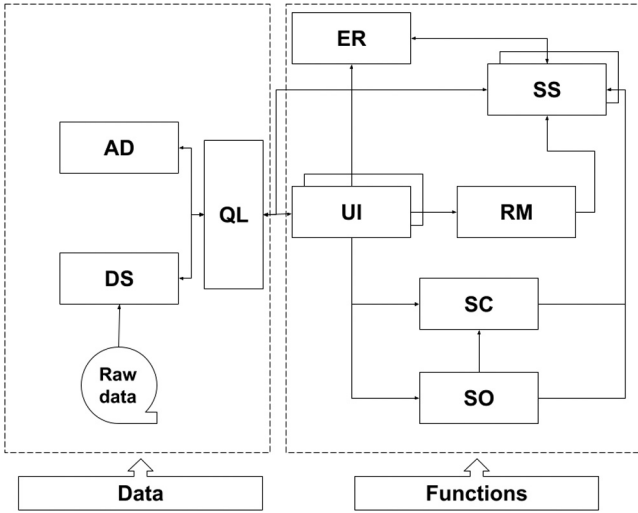


Fig. 3. Architecture sketch.

addressed by the implementation of the Data Storage and the Annotation Database; moreover as the Query Layer produces annotated documents in GateNLP, the consumer programs are not affected by data extensions. Extensibility of User Interfaces and services is guaranteed as they use the Query Layer to read and write the data, and because they are registered in the Service Catalog and run through the Request Manager and the Service Orchestrator. The requirement of **availability** is satisfied as, provided that the data related components (DS, AD and QL) are running, the User Interface(s) and Service Systems can be registered, run, stopped, changed and so on without interfering with each other. As they can be run on different computing units, they have no impacts on the performances of other components.

6. Experiments

With respect to the RQ1 we evaluate the proposed entity extraction and knowledge consolidation pipeline in the following experimental settings:

1. *Atomic*: we analyze the performance of each algorithm atomically to isolate it from the effects of error propagation. For NER we use GS_{NER} as the evaluation dataset and GS_{CON} for the knowledge consolidation tasks. We remind that knowledge consolidation tasks, i.e., NEL, NIL prediction, and NIL clustering are applied, and evaluated, only on annotations of type “person”, “location”, and “organization” (see Section 4). Moreover, This analysis aligns with the concept of human-in-the-loop validation, in which the results of each step can be refined by humans prior to being utilized as input for the subsequent stages within the pipeline. For NER, we additionally calculate performance metrics on a per-type basis and with different criteria, including more relaxed ones that also accept spans that are partially correct (see Section 6.1.1).
2. *NEL with NIL prediction*: we study the combined performance of NEL and NIL prediction, since these two tasks are closely related, isolating them from NER errors. To achieve this, we execute NEL and NIL prediction on gold standard NER annotations from GS_{CON} .
3. *End-to-end*: we analyze the behavior of our system on each mention of GS_{CON} for the tasks NER, NEL, and NIL prediction (as done for NEEL [59]), wherein an input mention can be finally linked to an entity in the KB or classified as NIL. NIL clustering is not included in the end-to-end evaluation because it complicates the evaluation since it would require evaluating

whether a NIL mention was grouped in the correct cluster, leading to a too broad space of possible outcomes (e.g., the NIL mention is grouped with one correct and one wrong mention, the NIL mention is grouped with only one correct mention out of n).

To better study the difficulty of the domain we compare with NER and NEL results on standard benchmarks in the Italian language: WikiNER [53] and I-CAB [60] for NER; Italian VoxEL [61] and NEEL-IT [59] for NEL.

To address RQ2, we study the domain-specific *extensibility* and *scalability* of our proposal. For the former, we consider the use case in which two rule-based additional extractors (RAE) are incorporated into the architecture to extract law articles and postal addresses. For the *scalability* experiment, we study the required time for processing a document as the number of words increases with the full pipeline and also with the two rule-based additional extractors.

In the remainder of this section, we present the evaluation metrics and the evaluation results of the experiments related to RQ1 and to RQ2.

6.1. RQ1 experiments

The different tasks involved in the RQ1 are analyzed according to the following metrics and criteria.

6.1.1. Evaluation metrics and criteria

For the *atomic* evaluation of NER, we calculate precision, recall, and F_1 score with different matching criteria that determine when an extracted mention is correct with respect to the human-annotated gold standard. Indeed in some contexts, “partially-correct” annotations might be acceptable, e.g., identifying “A4 Highway” instead of “A4 Highway Torino-Trieste” might be informative enough [1,62]. Furthermore, in a human-in-the-loop scenario, we prefer to detect partially-correct annotations rather than to miss them, since a human can quickly correct annotation boundaries.

The adopted criteria, taken from [62] and extended to provide both a typed and untyped evaluation, allow us to assess the behavior of the algorithms at different severity levels. The metrics are defined as follows:

1. *strong*: the predicted entity has an exact span match with the gold standard annotations;
2. *strong typed*: the predicted entity has an exact span match and the type is correct with respect to the gold standard annotations;
3. *approximate*: the predicted entity is contained in the correct span (or vice versa);
4. *approximate typed*: the predicted entity is contained in the correct span (or vice versa) and the type is correct;
5. *partial*: the predicted entity is overlapping with the correct span;
6. *partial typed*: the predicted entity is overlapping with the correct span and the type is correct;

For the sake of clarity, given c is the chosen criterion from the above, Y_c^i the number of correctly predicted annotations according to c , Y^i the total number of predicted annotations, and Y_G the expected number of annotations in the gold standard, precision is defined as $P = \frac{Y_c^i}{Y^i}$, that is the number of correctly predicted annotations divided by the total number of predicted annotations. Recall is $R = \frac{Y_c^i}{Y_G}$, the ratio of correctly predicted annotations and the expected number of annotations, and, finally F_1 score is the harmonic mean of P and R : $F_1 = 2 * \frac{P * R}{P + R}$.

NEL is *atomically* evaluated on accuracy, that is the number of mentions linked to the correct entity divided by the total number of mentions to link, and recall@k. This latter metric assesses whether the correct entity is among the top-k candidates with the highest linking score (in our case accuracy and recall@1 are equivalent). Note that

Table 3

Results obtained by our algorithms (SpaCy and Tint) on Italian benchmark datasets compared with recent competitive approaches. I-CAB [60] contains the types “person”, “location”, and “organization”. WikiNER [53] additionally considers the type “miscellaneous”.

	P	R	F ₁
SpaCy on WikiNER [53]	.919	.919	.919
GilBERTo ^a on WikiNER	.927	.927	.928
BERTino on WikiNER [63]	–	–	.904
BERTino Teacher Model on WikiNER [63]	–	–	.918
Tint on I-CAB [60]	.844	.800	.821

^a <https://github.com/idb-ita/GilBERTo>.

NEL, and also NIL prediction and NIL clustering, is evaluated solely on the types “person”, “location”, and “organization”.

For the *atomic* evaluation of the NIL prediction classifier, we calculate precision, recall, and F₁ score for both the NIL and ¬NIL classes. The NIL prediction task strongly relies on the NEL score, thus to evaluate it independently from NEL we consider correct when mentions linked to a wrong entity are classified as NIL: in this case, the top-ranked entity, according to NEL, is not the correct one and the NIL prediction classifier is based on the assumption that when the correct entity is not the top-ranked one, then, it does not exist in the KB.

NIL clustering, similarly to coreference resolution tasks [9], is evaluated *atomically* with the metrics MUC, B₃, and CEAF_c. For each of them, we calculate precision, recall, and F₁ score. For the evaluation to be independent of error propagation, we run the clustering algorithm only for the NIL mentions from the gold standard.

For the joint evaluation of *NEL with NIL prediction*, starting from the gold standard NER annotations, we calculate the accuracy on (i) all the mentions, (ii) on the mentions that should be linked to the KB, and (iii) on the mentions that should be classified as NIL. They respectively represent the ratio of all the mentions, of the ¬NIL mentions, and of the NIL mentions that were correctly processed.

Finally, for the *end-to-end* evaluation of NER, NEL, and NIL prediction, we expanded the evaluation criteria for NER defining *approximate linking* and *approximate typed linking* as the criteria that extend the respective NER metric by additionally considering NEL and NIL prediction. Thus an annotation is correct only when boundaries and type (when typed) are correct, and it is linked to the correct entity (and classified as ¬NIL) or correctly identified as NIL. For each of the two criterion, we measure precision, recall, and F₁ score.

The hyperparameters for the combination rules are determined with preliminary tests. The weights assigned to each algorithm are as follows: 0.3 for SpaCy, 0.1 for Tint, and 0.6 for TrieNER.

6.1.2. Results

First, we present a comparative table (Table 3) with the outcomes achieved by two of our NER algorithms, SpaCy (combined with BERT_{BASE} and Tint, across benchmark datasets. We also include the results from two other recent and competitive approaches for reference. It is visible that our SpaCy algorithm, which exploits a BERT encoder, is competitive with recent approaches.

In Table 4 we show the results for the *atomic* evaluation of NER. As expected there is a substantial difference between the results obtained using *strong* and *approximate* criteria, highlighting that our system in several cases is able to identify entities but struggles to find the best boundaries. However, boundary errors can be fixed relatively quickly with human-in-the-loop intervention. Furthermore, the difference between *approximate* and *partial* criteria is minimal, thus, we consider the *approximate* criteria permissive enough and we omit the *partial* criteria in the remainder of the section.

In Table 5 we report a per-type comparison of the *atomic* evaluation of NER algorithms and of their combination. The types “date”, “money”, and “miscellaneous” are recognized by a single algorithm, Tint for the former two and SpaCy for the latter. The different results

Table 4

Atomic NER evaluation (precision, recall, F₁ score) with different matching criteria (rows), typed and untyped. These results are obtained using the combination of the NER algorithms and considering all the evaluated types.

	Untyped			Typed		
	P	R	F ₁	P	R	F ₁
strong	.519	.437	.475	.447	.377	.409
approximate	.869	.680	.763	.646	.531	.583
partial	.874	.686	.769	.649	.533	.586

Table 5

Atomic NER evaluation by type and algorithm using *approximate typed match*. *Miscellanea mentions are excluded from the overall calculation. **Note that the type Organization is underrepresented in the knowledge base used by TrieNER, thus only a few entities of this type (i.e., 10) have been predicted.

Type	Algorithm	P	R	F ₁
Person	SpaCy	.921	.763	.835
	Tint	.903	.621	.736
	TrieNER	.767	.346	.477
	Combination	.815	.799	.807
Location	SpaCy	.354	.908	.510
	Tint	.492	.826	.617
	TrieNER	–	–	–
	Combination	.599	.601	.600
Organization	SpaCy	.582	.409	.481
	Tint	.656	.565	.607
	TrieNER	.900	.002	.005
	Combination	.342	.920	.499
Date	SpaCy	–	–	–
	Tint	.837	.552	.665
	TrieNER	–	–	–
	Combination	.837	.551	.665
Money	SpaCy	–	–	–
	Tint	.981	.573	.723
	TrieNER	–	–	–
	Combination	.981	.569	.720
Miscellaneous	SpaCy	.251	.041	.070
	Tint	–	–	–
	TrieNER	–	–	–
	Combination	.264	.039	.068
Overall*	SpaCy	.638	.432	.515
	Tint	.748	.601	.666
	TrieNER	.767	.113	.197
	Combination	.660	.676	.668

obtained by the combination may seem counterintuitive, but this difference is caused by boundary conflicts that can also happen between mentions of different types.

SpaCy is effective in recognizing the types “person” and “location” while Tint is superior for organizations. The evaluation of TrieNER obtains high precision and low recall. Indeed, this approach misses all the entities that are not included in its dictionary (in the experiments the dictionary is mostly composed of persons and a few organizations).

Our combination rules achieved a good compromise between precision and recall even if they do not outperform the best algorithm for each type: an F₁ score of 0.668 was obtained with the *approximate matching* criterion.

The results of SpaCy and Tint on standard benchmarks confirm the challenging and diverse nature of the domain: by comparing Tables 3 and 4 it is evident that NER algorithms experience a noticeable decline in performance when applied to domain-specific data. Looking at per-type performances (Table 5), across almost all types we observe lower performance compared to the results of the evaluation on benchmark datasets. Notably, in handling the class “person” our proposed system demonstrates relatively similar outcomes (F₁ > 0.8), while the identification of miscellanea mentions proves highly ineffective. This result can likely be attributed to the distinct characteristics of domain-specific miscellanea, which differ significantly from the miscellanea found in

Table 6

NEL results on benchmark datasets: strict (s-) and relaxed (r-) version of Italian VoxEL [61] and NEEL-IT@Evalita 2016 [59]. for NEEL-IT, Twitter profile tags (@username) and hashtags (#tag) were filtered out.

	s-VoxEL-it	r-VoxEL-it	NEEL-IT
Accuracy	.889	.647	.690
R@100	.968	.915	–

Table 7

Atomic evaluation of the knowledge consolidation tasks.

NEL				
	Accuracy	.735		
	R@100	.908		
NIL prediction				
	P	R	F ₁	
NIL	.922	.865	.892	
~NIL	.585	.720	.645	
NIL clustering				
	P	R	F ₁	
MUC	.719	.839	.774	
B ³	.164	.601	.258	
CEAF _e	.072	.317	.117	
NEL & NIL prediction				
	in KB	NIL	All	
Accuracy	.468	.919	.791	

Table 8

end-to-end NER, NEL, and NIL prediction evaluation.

	P	R	F ₁
approximate linking	.598	.689	.640
approximate typed linking	.523	.609	.563

standard benchmarks. For the remaining classes “location”, “organization”, “date”, and “money” the evaluation exhibits F₁ scores ranging between 0.5 and 0.7. This suggests that the tested algorithms, trained on the available benchmark datasets, have the capability to process these types, although there is still significant room for improvement, e.g., via domain-specific fine-tuning.

The results of the atomic evaluation of the different knowledge consolidation tasks are shown in Table 7. NEL achieves satisfying results, with an accuracy of 0.735, and a recall@100 of 90.8%, in line with the results obtained on benchmark dataset (see Table 6). With respect to NIL prediction, the classifier is effective in identifying NIL mentions but it suffers with ~NIL ones obtaining an F₁ score of 0.645. The biggest weakness of the consolidation process is the NIL clustering component which obtains low F₁ score values for B³ and CEAF_e metrics.

The joint evaluation of NEL and NIL prediction (independent from NER errors) results in the 79.1% of mentions correctly processed. These include NIL mentions that the system identified with a very high accuracy of 0.919, while for the mentions that should be linked to the KB, the accuracy is 0.468; this can be explained by a bias in the NIL prediction that tends to predict the NIL class.

Finally, Table 8 reports the evaluation with the end-to-end evaluation with the *approximate linking* and *approximate typed linking* criteria. The results obtained with *approximate linking*, which ignores the recognized NER type, are noticeably better than with *approximate typed linking*: the F₁ score is 0.640, almost 8% higher. This finding suggests that, once the mention has been linked by NEL, the linked entity may also allow to correct or consolidate the type identified by NER.

6.2. RQ2 experiments

This section presents the result from the *extensibility* and *scalability* experiments.

6.2.1. Extensibility

We develop the two rule-based additional extractors (RAE) using pattern-matching rules based on references to postal addresses and law

Table 9

Rule-based additional extractors (RAE) evaluation. The matching criterion is the strong typed matching.

Type	Algorithm	P	R	F ₁
Postal Address	RAE	.902	.881	.892
Law Articles	RAE	1.0	.702	.825

articles found in our corpus. In addition to identifying the boundaries of the mentions, the extractors also link postal addresses to geographic coordinates, using off-the-shelves geocoding services, and law articles to their pages on the Normattiva¹⁶ website, coherently with the example in Fig. 1.

We evaluate the two RAEs algorithms on the set of documents of GS_{CON} (see Section 3.1) annotated by a single annotator, which results in 210 annotations of law articles and 37 of postal addresses. We calculate precision, recall, and F₁ score with the strictest criterion *strong typed match*. The results are shown in Table 9.

“Postal address” and “law articles” can be considered as a specialization of “location” and “miscellaneous”. We observe that the F₁ score calculated for both these types are significantly higher than the ones calculated for “location” and “miscellaneous” (see Table 5). This confirms the hypothesis that combining rule-based entity extractors for entities mentioned with predictable patterns can be beneficial. Therefore, we integrate these annotations with the ones produced by the pipeline described in Section 4 by prioritizing RAEs’ annotations in case of overlap.

The results of the RAE performance evaluation are shown in Table 9. When comparing them with the NER results, it becomes evident that RAEs achieve significantly higher performance measures. This demonstrates the efficacy of rule-based algorithms for law articles and postal addresses and confirms the successful outcome of the integration process.

6.2.2. Scalability

In Fig. 4 we show the execution time of the full entity extraction and knowledge consolidation pipeline and of the RAEs. It is visible that time grows linearly with respect to the number of words processed. Note that the execution time of NIL clustering (part of the full pipeline) scales non-linearly with the number of NIL mentions to cluster, thus it is necessary to limit the document size and eventually divide long documents into smaller chunks. In our experiments, the algorithms are executed separately on each document that contains on average 2000 words.

6.3. Discussion

We now discuss the results in the context of the initial research questions, focusing on the limitations of the proposed solution and their impact on the specific application domain.

With respect to RQ1, in the *end-to-end* evaluation our system achieved an F₁ score of 0.640 (with the *approximate linking* criterion). This means that more than one mention out of two is correctly processed by NER, NEL, and NIL prediction. Taking into account the high specificity of the domain, the noisy input, and the absence of in-domain training or fine-tuning, we believe that the baseline for identifying and linking entities is promising. However, the final NIL clustering step achieved unsatisfactory results according to B³ and CEAF_e metrics, although MUC metrics seem satisfactory. Providing an intuitive interpretation of the clustering scores is more complicated. Results obtained by state-of-the-art NIL clustering algorithms on benchmark data in English vary significantly, across measures and datasets [9]. An ineffective NIL clustering step badly affects the

¹⁶ <https://www.normattiva.it/>

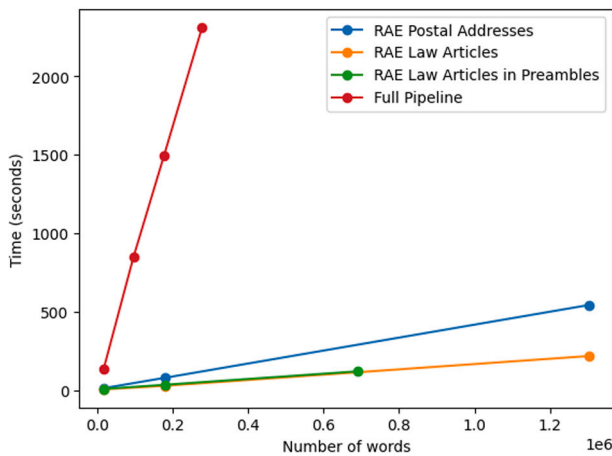


Fig. 4. Execution time of some analyses varying the number of processed words. Our point is to show the performance trend, not their absolute values, that are hardware dependent.

knowledge consolidation process of NIL mentions. A possible solution is to optimize the clustering hyperparameters (i.e., clustering thresholds) on a domain-specific dataset, indeed, recent work [9] suggests that in-domain optimization may be beneficial.

Also the observed bias of the NIL prediction algorithm towards the NIL class may be mitigated by domain-specific training. Nevertheless, it is important to note that within court judgments, this concern holds minor significance, given that the most relevant entities are usually NIL (e.g., plaintiff, defendants, judges, and attorneys).

The results also reveal that NER algorithms trained on accessible datasets and applied to the specific domain exhibit a reduction in effectiveness with the only exception of the type “person”. Considering that NER holds the first position within the pipeline, it significantly affects all the subsequent tasks. For this reason, we underscore the importance of domain-specific fine-tuning for the NER algorithms. We proved the impact of domain-specific fine-tuning in [64] in which we fine-tuned the SpaCy approach, which utilizes a BERT encoder, on our gold standard, achieving an F_1 score of 0.822, with the strictest *strong typed match* criterion. This increase by about 100% (see Table 4) has an impact also on the performance evaluated in the *end-to-end* experiment that with the domain-specific NER model achieves an F_1 of 0.661 with the *strong typed match* criterion (10% higher of the results calculated with *approximate typed* in Table 8).

As for RQ2, the rule-based additional extractors (RAE) and most of the steps in the full pipeline perform operations that scale linearly with the number of input words. The only exception is NIL clustering, which is sensitive to the number of mentions to cluster that depends on the number of words. However, applying the algorithms to bounded-size documents, and eventually dividing long documents into smaller chunks, allows the system to fulfill the requirement of scalability.

The implemented architecture meets scalability, extensibility, and availability requirements (see Section 5), enabling the processing of annotations for distinct document subsets and seamless expansion by adding nodes for increased document numbers. Indeed, several operations are performed by independent systems that can run asynchronously. These operations include the NER algorithms and the rule-based additional extractors, while other services like NEL strictly depend on the previous steps. As a consequence, the architecture is compatible with the *map reduce* strategy, allowing the repository of documents to be efficiently loaded and analyzed in stages, and eliminating the need for an expensive *cold start* process.

The results discussed so far provide insights into the feasibility, in terms of effectiveness and efficiency, of a solution that extracts and semantically consolidates entity mentions from Italian judgments.

Our architecture additionally offers the capability of human-in-the-loop revision of the annotations produced by the algorithms. Furthermore, the human-in-the-loop process we used for creating the gold standard enabled us to estimate 30 min as the average required time to fully revise the automatic annotations of an Italian judgment. This translates to the revision of 320 judgments per month by a single full-time annotator. This estimation serves as an upper limit for assessing the costs linked to human-involved entity extraction.

7. Conclusion and future works

In our paper, we proposed an entity-centric framework designed for the effective management of legal documents, particularly court judgments. This framework revolves around structuring a repository of textual documents and enhancing their utility through meticulous annotation. These annotations cater to a diverse range of subsequent tasks. The documents undergo preliminary processing before undergoing iterative annotation. This annotation process is facilitated by a set of NLP services that synergistically combine machine learning and rule-based strategies to ensure comprehensive coverage and accuracy.

The framework is designed to be developed and maintained in a sustainable way, allowing for multiple services and uses of the annotated document repository. The scarcity of annotated data was considered an intrinsic challenge for its development. This design activity is the result of a cooperative project where a scientific team, institutional bodies, and companies appointed to implement the final system were involved in co-design activities.

In the second part of the paper, we described experiments to demonstrate the feasibility of the solution and the main challenges to scaling the system at a national level. In particular, the results obtained in annotating data with different low-resource methods and with solutions designed to combine these approaches in a meaningful way were reported. An essential aspect of the proposed solution is a human-in-the-loop approach to control the output of the annotation algorithms in agreement with the organizational processes in place in Italian courts. Based on these results, the feasibility of the proposed approach was advocated and the challenges that must be addressed to ensure the scalability and robustness of the proposed solution were discussed.

As part of future work, we plan to further explore the human-in-the-loop integration by incorporating mechanisms that learn from user feedback, similar to what has been done for other tasks (e.g., knowledge exploration [65], ontology matching [66], entity reconciliation [67]).

We observe that while our experiments are focused on a national case study, the system can be adapted to handle legal documents from different countries and languages with minor adjustments. Specifically, only the Service Systems need replacement, as they employ language-specific models and rules, and the Entity Registry metamodel should be reviewed to accommodate country-specific codes and conventions for identifying persons, laws, etc. Furthermore, after a thorough entity type revision, the system could handle documents related to other subjects.

Declaration of competing interest

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

Data availability

The authors do not have permission to share data.

Acknowledgments

This research has received funding from:

- CINI (Consorzio Interuniversitario Nazionale per l'Informatica), in the context of the project Datalake@Giustizia;
- the Italian Ministry of Justice, in the context of the project PON Next Generation UPP;
- the European Union's Horizon Europe research and innovation programme (grant agreement No 101070284 - enRichMyData);
- the Università degli Studi di Milano within the program "Piano di sostegno alla ricerca";
- the European Union within the MUSA – Multilayered Urban Sustainability Action – project, in the NextGenerationEU National Recovery and Resilience Plan (NRRP) Mission 4 Component 2 Investment Line 1.5: Strengthening of research structures and creation of R&D "innovation ecosystems", set up of "territorial leaders in R&D";
- the European Union within the SERICS projects (PE00000014) under the MUR NRRP - NextGenerationEU.

References

- Nadeau D, Sekine S. A survey of named entity recognition and classification. *Linguisticae Investigationes* 2007;30(1).
- Armentano MG, Godoy D, Campo M, Amandi A. NLP-based faceted search: Experience in the development of a science and technology search engine. *Expert Syst Appl* 2014;41(6):2886–96.
- Çetindağ C, Yazıcıoğlu B, Koç A. Named-entity recognition in Turkish legal texts. *Nat Lang Eng* 2023;29(3).
- Castano S, Falduti M, Ferrara A, Montanelli S. A knowledge-centered framework for exploration and retrieval of legal documents. *Inf Syst* 2022;106.
- Pozzi R, Moiraghi F, Lodi F, Palmonari M. Evaluation of incremental entity extraction with background knowledge and entity linking. In: *Proceedings of the 11th international joint conference on knowledge graphs. IJCKG '22*, New York, NY, USA: ACM; 2023.
- Kassner N, Petroni F, Plekhanov M, Riedel S, Cancedda N. EDIN: An end-to-end benchmark and pipeline for unknown entity discovery and indexing. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics*; 2022, p. 8659–73.
- Heist N, Paulheim H. NASTyLinker: NIL-aware scalable transformer-based entity linker. In: *The Semantic Web, ESWC 2023. Lecture Notes in Computer Science. Springer*; 2023, p. 174–91.
- McNamee P, Dang HT. Overview of the TAC 2009 knowledge base population track. In: *Second text analysis conference (TAC 2009)*. Vol. 2, 2009.
- Logan IV RL, McCallum A, Singh S, Bikel D. Benchmarking scalable methods for streaming cross-document entity coreference. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics; 2021, p. 4717–31.
- Wu L, Petroni F, Josifoski M, Riedel S, Zettlemoyer L. Scalable zero-shot entity linking with dense entity retrieval. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics; 2020, p. 6397–407.
- Sevgili O, Shelmanov A, Arkhipov MV, Panchenko A, Biemann C. Neural entity linking: A survey of models based on deep learning. *Semantic Web* 2020;13.
- Wang X, Jiang Y, Bach N, Wang T, Huang Z, Huang F, et al. Automated concatenation of embeddings for structured prediction. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021, p. 2643–60.
- Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics; 2019, p. 4171–86.
- Paccosi T, Palmero Aprosio A. KIND: an Italian multi-domain dataset for named entity recognition. In: *Proceedings of the thirteenth language resources and evaluation conference (LREC)*. Marseille, France: European Language Resources Association; 2022, p. 501–7.
- He Z, Liu S, Li M, Zhou M, Zhang L, Wang H. Learning entity representation for entity disambiguation. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics; 2013, p. 30–4.
- Yamada I, Shindo H, Takeda H, Takefuji Y. Joint learning of the embedding of words and entities for named entity disambiguation. In: *Proceedings of the 20th SIGNLL conference on computational natural language learning. Association for Computational Linguistics*; 2016.
- Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: *Advances in neural information processing systems*. Vol. 30, Curran Associates, Inc.; 2017.
- De Cao N, Wu L, Papat K, Artetxe M, Goyal N, Plekhanov M, et al. Multilingual autoregressive entity linking. *Trans Assoc Comput Linguist* 2022;10.
- Procopio L, Conia S, Barba E, Navigli R. Entity disambiguation with entity definitions. In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics*; 2023, p. 1297–303.
- Xu M, Nosirova N, Jiang K, Wei F, Jiang H. FOFE-based deep neural networks for entity discovery and linking. In: *Text analysis conference (TAC 2017)*. Vol. 10, 2017.
- Blissett K, Ji H. Cross-lingual NIL entity clustering for low-resource languages. In: *Proceedings of the second workshop on computational models of reference, anaphora and coreference*. Minneapolis, USA: Association for Computational Linguistics; 2019.
- Zirlikly A, Diab MT, Benajiba Y. GWU english TAC-KBP EL diagnostic task with name mention. In: *Text analysis conference (TAC 2015)*. Vol. 8, 2015.
- Zhong H, Xiao C, Tu C, Zhang T, Liu Z, Sun M. How does NLP benefit legal system: A summary of legal artificial intelligence. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics*; 2020, p. 5218–30.
- Kanapala A, Jannu S, Pamula R. Passage-based text summarization for legal information retrieval. *Arab J Sci Eng* 2019;44.
- Carvalho D, Tran V, Tran V-K, Minh L-N. Improving legal information retrieval by distributional composition with term order probabilities. In: *4th Competition on Legal Information Extraction and Entailment (COLIEE 2017)*, 16th International Conference on Artificial Intelligence and Law (ICAIL 2017). 2017.
- Perotto F, Verstaavel N, Trabelsi I, Vercouter L. Combining bandits and lexical analysis for document retrieval in a juridical corpora. In: *Artificial Intelligence XXXVII: 40th SGA International Conference on Artificial Intelligence, AI 2020*, Cambridge, UK, December 15–17, 2020, *Proceedings*. Springer; 2020, p. 317–30.
- Dragonì M, Villata S, Rizzi W, Governatori G. Combining natural language processing approaches for rule extraction from legal documents. In: *AICOL 2017: AI approaches to the complexity of legal systems, lecture notes in computer science*. Vol. 10791, 2018, p. 287–300.
- Kejriwal M, Szekely P, Knoblock C. Investigative knowledge discovery for combating illicit activities. *IEEE Intell Syst* 2018;33.
- Pérez FJ, Garrido VJ, García A, Zambrano M, Kozik R, Choraś M, et al. Multimedia analysis platform for crime prevention and investigation. *Multimedia Tools Appl* 2021.
- Batini C, Bellandi V, Ceravolo P, Moiraghi F, Palmonari M, Siccardi S. Semantic data integration for investigations: Lessons learned and open challenges. In: *2021 IEEE international conference on smart data services (SMDS)*. 2021.
- Sarika J, Pooja H, Nandana M, Sudipto G, Abhinav D, Ankush B. Constructing a knowledge graph from Indian legal domain corpus. In: *Text2KG 2022: International workshop on knowledge graph generation from text, co-located with the ESWC 2022*. Vol. 3184, 2022.
- Andrew JJ, Tannier X. Automatic extraction of entities and relation from legal documents. In: *Proceedings of the Seventh Named Entities Workshop. Association for Computational Linguistics*; 2018, p. 1–8.
- Leitner E, Rehm G, Moreno-Schneider J. Fine-grained named entity recognition in legal documents. In: *Semantic systems. the power of AI and knowledge graphs: 15th international conference, SEMANTICS 2019. Springer*; 2019, p. 272–87.
- Huang J, Meng Y, Han J. Few-shot fine-grained entity typing with automatic label interpretation and instance generation. In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. KDD '22*, 2022, p. 605–14.
- Cardellino C, Teruel M, Alemany LA, Villata S. A low-cost, high-coverage legal named entity recognizer, classifier and linker. In: *Proceedings of the 16th edition of the international conference on artificial intelligence and law. ICAIL '17, Association for Computing Machinery*; 2017, p. 9–18.
- 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. AICCC '18, Association for Computing Machinery*; 2018, p. 23–8.
- Tamper M, Oksanen A, Tuominen J, Hietanen A, Hyvönen E. Automatic annotation service APPI: Named entity linking in legal domain. In: *The semantic web: ESWC 2020 satellite events. ESWC 2020. Lecture Notes in Computer Science. Springer*; 2020, p. 208–13.
- Klie J-C, Eckart de Castilho R, Gurevych I. From zero to hero: Human-in-the-loop entity linking in low resource domains. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics*; 2020, p. 6982–93.
- Humphreys L, Boella G, van der Torre L, et al. Populating legal ontologies using semantic role labeling. *Artif Intell Law* 2021;29:171–211.

- [40] Hu W, Zhao S, Zhao Q, Sun H, Hu X, Guo R, et al. BERT_{LF}: A similar case retrieval method based on legal facts. *Wirel Commun Mob Comput* 2022;2022.
- [41] Rabelo J, Goebel R, Kim M-Y, et al. Overview and discussion of the competition on legal information extraction/entailment (COLIEE) 2021. *Rev Socionetwork Strateg* 2022;16.
- [42] Pauzi Z, Capiluppi A. Applications of natural language processing in software traceability: A systematic mapping study. *J Syst Softw* 2023;198.
- [43] Breit A, Waltersdorfer L, Ekaputra FJ, Sabou M. An architecture for extracting key elements from legal permits. In: 2020 IEEE international conference on big data (big data). 2020, p. 2105–10.
- [44] Amato F, Mazzeo A, Penta A, Picariello A. Using NLP and ontologies for notary document management systems. In: Database and expert systems application, 2008. DEXA'08. 2008, p. 67–71.
- [45] Buey MG, Garrido AL, Bobed C, Ilarri S. The AIS project: Boosting information extraction from legal documents by using ontologies. In: Proceedings of the 8th International Conference on Agents and Artificial Intelligence (ICAART 2016). Vol. 2, 2016, p. 438–45.
- [46] Ruiz M, Roman C, Garrido AL, Mena E. uAIS: An experience of increasing performance of NLP information extraction tasks from legal documents in an electronic document management system. In: Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020). Vol. 1, 2020, p. 189–96.
- [47] Cugno M, Giacomelli S, Malgieri L, Mocetti S, Giuliana Palumbo - Banca d'Italia. La giustizia civile in Italia: durata dei processi, produttività degli uffici e stabilità delle decisioni, in *Questioni di Economia e Finanza*. 2023.
- [48] Jha K, Röder M, Ngonga Ngomo A-C. All that glitters is not gold—rule-based curation of reference datasets for named entity recognition and entity linking. In: The Semantic Web. ESWC 2017. Lecture Notes in Computer Science. Springer; 2017.
- [49] Braylan A, Alonso O, Lease M. Measuring annotator agreement generally across complex structured, multi-object, and free-text annotation tasks. In: Proceedings of the ACM web conference 2022. 2022, p. 1720–30.
- [50] Deleger L, Li Q, Lingren T, Kaiser M, Molnar K, Stoutenborough L, et al. Building gold standard corpora for medical natural language processing tasks. In: AMIA annual symposium proceedings. American Medical Informatics Association; 2012.
- [51] Hripcsak G, Rothschild AS. Agreement, the f-measure, and reliability in information retrieval. *J Am Med Inform Assoc* 2005;12(3).
- [52] Grouin C, Rosset S, Zweigenbaum P, Fort K, Galibert O, Quintard L. Proposal for an extension of traditional named entities: From guidelines to evaluation, an overview. In: Proceedings of the 5th Linguistic Annotation Workshop. 2011, p. 92–100.
- [53] Nothman J, Ringland N, Radford W, Murphy T, Curran JR. Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence* 2013;194:151–75.
- [54] Palmero Aprosio A, Moretti G. Tint 2.0: an all-inclusive suite for NLP in Italian. In: Proceedings of the Fifth Italian Conference on Computational Linguistics CLiC-it 2018, Vol. 10. 2018, p. 311–7.
- [55] Manning CD, Surdeanu M, Bauer J, Finkel J, Bethard SJ, McClosky D. The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 2014, p. 55–60.
- [56] Monahan S, Lehmann J, Nyberg T, Plymale J, Jung A. Cross-lingual cross-document coreference with entity linking. In: Text analysis conference (TAC 2011). 2011.
- [57] Cunningham H, Maynard D, Bontcheva K, Tablan V. GATE: an architecture for development of robust HLT applications. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics; 2002, p. 168–75.
- [58] Bellandi V, Siccardi S. An entity registry model. In: 4th International Conference on Natural Language Processing, Information Retrieval and AI (NIAI 2023). 2023.
- [59] Basile P, Caputo A, Gentile AL, Rizzo G. Overview of the evalita 2016 named entity recognition and linking in italian tweets (neel-it) task. In: EVALITA. Evaluation of NLP and Speech Tools for Italian: Proceedings of the Final Workshop 7 December 2016. 2016, p. 40–7.
- [60] Magnini B, Pianta E, Girardi C, Negri M, Romano L, Speranza M, et al. I-CAB: the Italian content annotation bank. In: Proceedings of the fifth international conference on language resources and evaluation (LREC'06). European Language Resources Association (ELRA); 2006.
- [61] Rosales-Méndez H, Hogan A, Poblete B. VoxEL: a benchmark dataset for multilingual entity linking. In: The Semantic Web. ISWC 2018. ISWC 2018. Lecture Notes in Computer Science. Springer; 2018.
- [62] Tsai RT-H, Wu S-H, Chou W-C, Lin Y-C, He D, Hsiang J, et al. Various criteria in the evaluation of biomedical named entity recognition. *BMC Bioinform* 2006;7.
- [63] Muffo M, Bertino E. BERTino: an Italian DistilBERT model. 2023, arXiv:2303.18121.
- [64] Pozzi R, Rubini R, Bernasconi C, Palmonari M. Named Entity Recognition and Linking for Entity Extraction from Italian Civil Judgements. In: *AlxIA 2023 – Advances in Artificial Intelligence: 22nd International Conference of the Italian Association for Artificial Intelligence, AlxIA 2023, Rome, Italy, November 6 – 9, 2023, Proceedings*, Springer (Forthcoming).
- [65] Bianchi F, Palmonari M, Cremaschi M, Fersini E. Actively learning to rank semantic associations for personalized contextual exploration of knowledge graphs. In: The Semantic Web. ESWC 2017. Lecture Notes in Computer Science. Springer; 2017, p. 120–35.
- [66] Cruz IF, Loprete F, Palmonari M, Stroe C, Taheri A. Pay-as-you-go multi-user feedback model for ontology matching. In: Knowledge Engineering and Knowledge Management. EKAW 2014. Lecture Notes in Computer Science. Springer; 2014, p. 80–96.
- [67] Li G. Human-in-the-loop data integration. *Proc. VLDB Endow.* 2017;10(12):2006–17.