Semi-automatic knowledge population in a legal document management system

Noname manuscript No.
(will be inserted by the editor)

Semi-Automatic Knowledge Population in a Legal Document Management System

Guido Boella · Luigi Di Caro · Valentina Leone

the date of receipt and acceptance should be inserted later

Abstract Every organization has to deal with operational risks, arising from the execution of a company's primary business functions. In this paper, we describe a legal knowledge management system which helps users understand the meaning of legislative text and the relationship between norms. While much of the knowledge requires the input of legal experts, we focus in this article on NLP applications that semi-automate essential time-consuming and lower-skill tasks - classifying legal documents, identifying cross-references and legislative amendments, linking legal terms to the most relevant definitions, and extracting key elements of legal provisions to facilitate clarity and advanced search options. The use of Natural Language Processing tools to semi-automate such tasks makes the proposal a realistic commercial prospect as it helps keep costs down while allowing greater coverage.

1 Introduction

Every organization has to deal with operational risks, arising from the execution of a company's primary business functions. Operational risks include monetary loss, fraud, physical or environmental risks, risks related to human resources, regulatory compliance and so forth. Risk Management departments typically collect and assess data for each risk in order to make management decisions, increasingly using ICT support. For one type of risk there is a lack of

Guido Boella
Department of Computer Science, University of Turin
E-mail: boella@di.unito.it

Luigi Di Caro
Department of Computer Science, University of Turin
E-mail: dicaro@di.unito.it

Valentina Leone
Department of Computer Science, University of Turin
E-mail: valentina.leone@unito.it

(Article begins on next page)

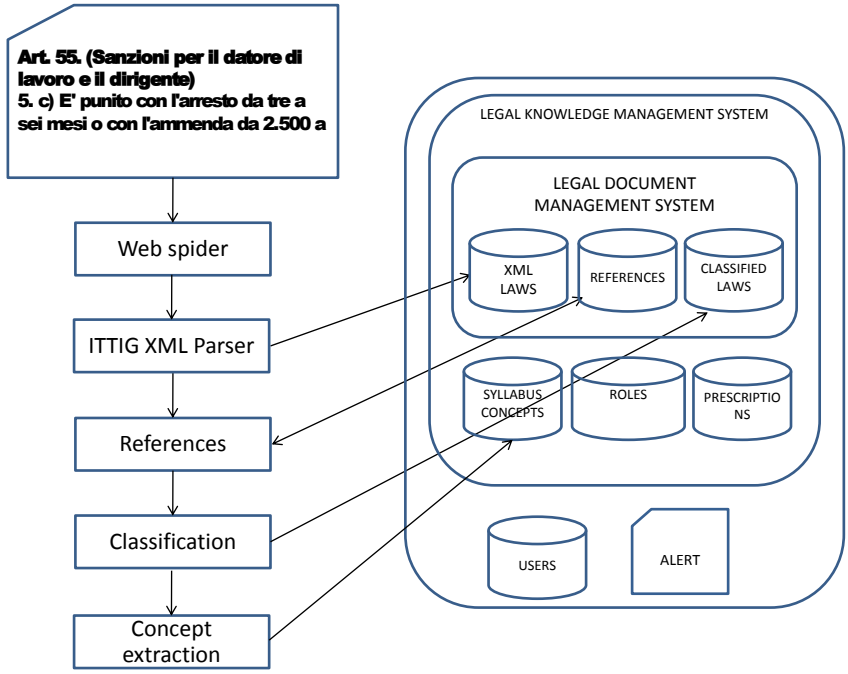

Fig. 1 The architecture of the proposed system.

3.2 Ontology

The Eunomos system incorporates Legal Taxonomy Syllabus [2,1], a specialist multilevel multilingual ontology, and extends the LTS framework to include an ontology of prescriptions. The LTS ontology of terms is used to explain the meaning of terms of art to users. The ontology of prescriptions is used to explain norms and all their components. Eunomos also incorporates the well known Eurovoc Thesaurus - a multilingual, multidisciplinary thesaurus with about 7,000 categories covering the activities of the EU. This is used as an independent means to classify documents (as in [6]).

One important and distinctive feature of LTS is that it allows for the fact that legal terms can mean different things in different contexts. To properly manage terminological and conceptual misalignment, a distinction is made between *legal terms* and *legal concepts*. The Legal Taxonomy Syllabus ontology framework stores concepts and terms in separate database tables. Legal terms can be single words or sets of words. It is possible for the same term to be related to multiple concepts, possibly in different domains. The original LTS ontology framework has been extended to model not only definitions of legal terms but also to describe prescriptions, taking inspiration from the way compliance officers in financial institutions extract norms for regulatory com-

pliance purposes. The prescription is defined as a concept which is necessarily related to the following concepts:

- *Deontic clause*: the type of prescription, i.e., obligation, prohibition, permission, exception.
- *Active role*: the addressee of the norm (e.g., citizen, director).
- *Passive role*: the beneficiary of the norm.
- *Crime*: the type of crime which occurs when the prescription is not adhered to (if it is an obligation or prohibition).
- *Sanction*: the concept describing the sanction resulting from the violation.

Both concepts and prescriptions are associated with a textual description and with the relevant legal sources: legislation via URN associations and case law via text quotation. For each prescription instance, the interpretation of relevant texts is explained in natural language. The prescription instance also links to in-text references to other articles and concepts defined in the ontology. For instance, the relevant fields for active role (e.g. director), passive role (e.g. consumer), and sanction are all defined within the ontology and are linked to from the prescription, as well as concepts occurring in the description.

Structuring prescriptions in this way enables the user to make fine-tuned searches such as 'List the prescriptions for which the director concept has the active role', a useful feature for a compliance officer, particularly as legislation is typically not structured in a way that clearly delineates individual prescriptions laying out all the constituent elements. Self-contained prescriptions within legislation can span several paragraphs and/or articles; conversely a single paragraph within one article can include more than one prescription. Moreover, some elements of prescriptions can be found in other legislation such as the Penal Code. Another aspect to consider is that legislation often contains general principles from which several prescriptions are derived. The ontology of prescriptions thus allows a macro-prescription to be stored which specifies a general principle and contains links to specific prescriptions that come under this principle. The identification of the concepts related to a prescription is supported by a tool for semi-automated concept and relation extraction.

Language	National	European
French	8	47
Italian	28	52
English	71	75
Spanish	41	60
German	66	98
total	214	332

Number of current terms in ELTS

Language	National	European
French	7	43
Italian	24	45
English	54	71
Spanish	34	56
German	52	75
total	171	290

Number of current concepts in ELTS

Fig. 2 Terms and concepts in the ontology.

The terms of the ontology were initially extracted from a corpus of 24 EC directives, and 2 EC regulations. Occurrences of such entries were detected

from national transposition laws from English, French, Spanish, Italian and German jurisdictions. The actual number of annotated terms and concepts are provided in Figure 2.

3.3 The web interface

Figure 3 shows the web interface for searching prescriptions (partly translated into English to aid the reader's understanding). The figure shows a prescription displayed in the web browser concerning the obligation to inform workers about risks (see the red circled 1). The prescription is associated with a textual description, whose terms are linked to the concepts in the ontology (links in blue font). Thanks to a popup window, the definition of the term "Employer" is shown (2), derived from the Syllabus ontology. The prescription itself is a special concept in the ontology associated with other concepts (3), e.g., the concept "Director" plays the "active role" and the "Worker" the "passive role". The prescription is classified under the topic "Information". Below the description the legislative sources are shown. The link between the prescription in the ontology and the database of laws is made via legislative XML unique identifiers (URN). References are shown as hyperlinks, and again a popup window (6) is used to show the content of the referred norm (Art. 55, item 5, letter c). Popups are generated dynamically from the legislation database using the URN specified in the reference.

4 NLP pipeline

The proposed system processes legal documents using Natural Language Processing (NLP) techniques. In this section, we illustrate the entire NLP framework and pipeline which underlies all our semantic tools for semi-automating knowledge population.

4.1 The TULE parser

We defined rule-based procedures, drawn from the ones used in [35] and [36], for extracting and linking entities from the documents and statistical procedures for classifying the documents and extracting concepts and relations. These procedures will be described in more details in the following sections.

Both rule-based and statistical procedures take as input the result of the TULE parser [26], an open-source rule-based dependency parser for Italian and English developed at the Department of Computer Science of the University of Turin. The knowledge bases of TULE have been extended and updated for more than twenty years, therefore the accuracy of the TULE parser is strong and has one of best attested performance for Italian ([9]). The parser has been used successfully in several research and industry projects.

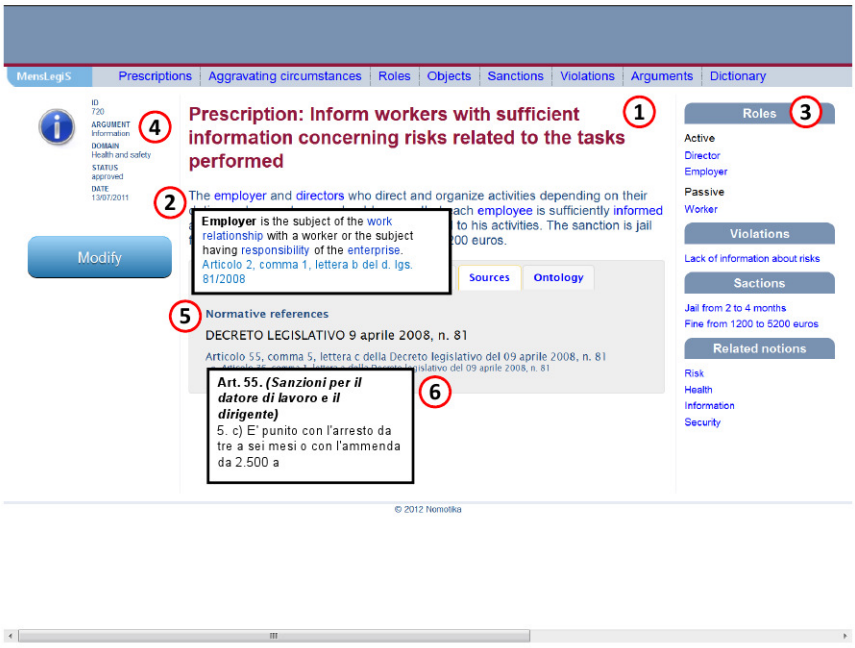

Fig. 3 An example of prescription.

The TULE parser establishes syntactic dependency relations, such as `SUBJ`, `OBJ`, etc., among pairs of words. In a dependency relation, we may identify a dominant word (the head) and a dominated word (the dependent). In the case of `SUBJ`, the head is a verb, while the dependent is a noun or a determiner that heads the sub-tree including the words in the subject.
Before building the syntactic dependencies, the TULE parser analyzes the words morphologically and disambiguates in the case of multiple morphological analyses (POS-tagging). Dependencies are established on the basis of the POS-tagger's result. Both the result of the POS-tagger and the syntactic dependencies are returned in the output, which is in textual format.

For instance, the Italian sentence "Così deciso in Roma nella camera di consiglio del giorno 12 marzo 2014" (Thus it was decided in Rome in the council chamber on 12 March 2014) results in the following TULE analysis:

```
1 Così (COSÌ ADV MANNER) [2;ADVB-RMOD]
2 deciso (DECIDERE VERB MAIN PARTICIPLE PAST M S) [0;TOP-VERB]
3 in (IN PREP MONO) [2;PREP-RMOD]
4 Roma (NOUN PROPER CITY) [3;PREP-ARG]
5 nella (IN PREP MONO) [2;PREP-RMOD]
5.1 nella (IL ART DEF F S) [5;PREP-ARG]
6 camera (CAMERA NOUN COMMON F S) [5.1;DET+DEF-ARG]
7 di (DI PREP MONO) [6;PREP-RMOD]
8 consiglio (CONSIGLIO NOUN COMMON F M) [6;DET+DEF-ARG]
```

the keyword match the morphological descriptions precisely and in the same order $Morph_{W_{n1}}$, ... , $Morph_{W_{nx}}$ and whether the preceding words similarly match the morphological descriptions $Morph_{W_{p1}}$, ... , $Morph_{W_{py}}$.
$dist_{n1}$, ... , $dist_{nx}$, $dist_{p1}$, ... , $dist_{py}$ are integers specifying the maximal distance among a pair of words. For instance, between the keyword and the word W_{n1} there could be a $dist_{n1}$ of other words.

If the three checks are satisfied, the rule is satisfied. In that case, the rule system takes some actions depending on certain attributes specified in the rule, which in turn depend on the specific task the rule is used for.

A concrete example of a pattern-matching rule is shown below. The rule recognizes the pattern "direttore di banca" (bank director) and all its morphological variants.

```
<rule>
   <headAlternatives>
      <head>
         <Lemma>direttore</Lemma>
         <Pos>Noun</Pos>
      </head>
   </headAlternatives>
   <nextAlternatives>
      <next maxDistance="1">
         <headAlternatives>
            <head>
               <Lemma>di</Lemma>
               <Pos>Preposition</Pos>
            </head>
         </headAlternatives>
         <nextAlternatives>
            <next maxDistance="2">
               <headAlternatives>
                  <head>
                     <Lemma>banca</Lemma>
                     <Pos>Noun</Pos>
                  </head>
               </headAlternatives>
            </next>
         </nextAlternatives>
      </next>
   </nextAlternatives>
</rule>
```

The rule is triggered with every occurrence of "direttore" as lemma. It is satisfied if a word corresponding to "direttore" as lemma is immediately (`maxDistance="1"`) followed by a word having "di" as lemma, and the latter is in turn followed, after at most two words (`maxDistance="2"`), by a word having "banca" as lemma. For instance, a linguistic variant of "direttore di banca" recognized by the rule is "direttrici delle banche". Note that "delle" is a compound, so that it counts as two words.

The results of the pattern-matching tool are used only as "suggestions" to the human annotator, in order to facilitate and speed-up his work.

4.3 Statistical Framework

Eunomos also contains a framework for the computation of statistical information over text collections and conceptual descriptors organized in ontological structures. In general, words and metadata information follow a process of numerical transformation which renders the textual documents computationally usable for unsupervised tasks like indexing, retrieval and comparisons as well as for supervised tasks.

4.3.1 Unupervised module

The process of transforming text into vectors requires the selection of suitable terms, and use of a weighting function as part of the frequency calculations. We use the *Term Frequency-Inverse Document Frequency* (TF-IDF) weighting function as proposed in [38], in order to take into account both the frequency of a term in a text and how it characterizes the text itself among the others. There are pre-processing steps that can be carried out on the selection and transformation of terms, which have been shown to be more effective than a simple bag-of-words approach. A commonly-accepted technique is to use a stopword list to remove typically uninformative terms and morphological transformation to reduce linguistic variability, transforming all terms to their lexical roots (i.e., the lemmas). The aim of these procedures is to eliminate noise while collapsing semantics. Typically, only nouns are left to be considered. The accuracy of the classification methods of using lists of stopwords and external resources such as WordNet [29] to extract the lemmas is highly dependent on the quality of these procedures. The problem is that WordNet-like methods which only have top-domain ontologies are unable to recognize and lemmatize many legal domain-specific terms. We therefore have to use a more complex approach - we use a dependency parser for Italian called TULE [27] that performs a deep analysis over the syntactic structure of the sentences and allows a direct selection of the informative units, i.e., the lemmatized nouns.

Eunomos also uses a text similarity algorithm, the Cosine Similarity, to find the most similar pieces of legislation in the whole database. The Cosine Similarity metric uses the TF-IDF measure to gauge the relative weight to be apportioned to various key words in the respective documents.

4.3.2 Supervised module

The module includes well-known Machine Learning mothods for automatic classification tasks. Eunomos makes use of Support Vector Machines (SVM), since it usually achieves high accuracy levels for textual data [13]. SVM makes use of the vectorial representation of the texts [39] and works by calculating the hyperplane having the maximun distance with respect to the nearest data examples. More in detail, we used *Liblinear* [20], a library for linear classification that is suited for fast text classification tasks on large datasets. In fact,

```
<articolo id="art1" xml:lang="it">
 <inlinemeta>
  <disposizioni>
   <modificheattive>
    <dsp:sostituzione implicita="no">
     <dsp:pos xlink:href="#art1-com1" xlink:type="simple" />
     <dsp:norma
       xlink:href="urn:nir:stato:regio.decreto:1942-03-16;267:legge.fallimentare">
       <dsp:pos xlink:href="#rif8"/>
     </dsp:norma>
     <dsp:novella><dsp:pos xlink:href="#mod185-vir1"/></dsp:novella>
    </dsp:sostituzione>
   </modificheattive>
  </disposizioni>
 </inlinemeta>
 <num>Art. 1.</num>
 <rubrica xml:lang="it"> Sostituzione dell'
  <rif id="rif7"
   xlink:href="urn:nir:stato:regio.decreto:1942-03-16;267:legge.fallimentare#art1">
   articolo 1 del regio decreto 16 marzo 1942, n. 267 </rif>
 </rubrica>
```

Fig. 5 An example of NIR XML annotation.

SVM-based classifiers usually have limitations on the size of the input data, while Liblinear can work on data with millions of instances and features.

4.4 Legislative XML

The system converts legislation into NormaInRete (NIR) XML format using the Institute of Legal Information Theory and Techniques (ITTIG)'s XML parser[3] if they are in pure textual format.[4] Maintaining laws in NIR XML format makes it easier for the system to extract elements such as paragraphs, articles and references so that knowledge engineers can categorize and annotate the elements, and lawyers can view specific relevant information. Within the Eunomos database, the unique identifier for each legislation and elements within legislation is the URN. URNs facilitate the construction of a global hypertext among the legal documents in a network environment with computer resources distributed among several publishers. It also allows the construction of knowledge bases containing the relationships between these documents.

A URN can be used in an XML or HTML file, e.g.:

```
<urn valore="urn:nir:stato:legge:1996-12-31;675"/>
```

The segment of Figure 5 shows an article which modifies existing legislation. The URN address of the modified legislation is provided in the header section denoted by the <inlinemeta> tag. We have included a small part of

[3] www.xmleges.org

[4] The Arianna portal already exports documents to NIR XML format.

the article to show the references to the URN addresses being used within the article text.

Eunomos uses the XML Leges Linker tool developed by ITTIG to find cross-references, an URN name resolver to obtain actual addresses of legislative articles, and XSLT to find and display outgoing and incoming hypertext links.

5 NLP applications

5.1 Classification of norms in accordance with the Eurovoc Thesaurus

Classification of legal text is an important task given the large amount of documents to be kept in specific contexts, and the possible risk at missing relevant information. In fact, this process usually involves intensive manual work which is slow and costly. Knowledge engineers specify the domain to which each norm belongs, selecting key terms within a domain-specific ontology. Given the amount of legal text documents produced every day and the huge mass of pre-existing documents to be classified, high-quality automated or semi-automated classification methods are welcome in this domain. In this section, we present our approach for the automatic classification of multi-label legal documents in accordance with the Eurovoc Thesaurus.

5.1.1 Data

For our experiments, we used JRC-Acquis-it[5], a freely-available parallel corpus of around 20,000 legislative text documents written from the 1950s onwards. Most of these documents were manually labelled in accordance with the Eurovoc thesaurus. The dataset JRC-Acquis has been already been used in [40], and it is known to contain very skewed data, which makes it difficult to learn models.

5.1.2 Pre-processing

The process of transforming text into vectors requires selection of suitable terms, and the use of a weighting function as part of frequency calculations. The accuracy of the classification methods is highly dependent on the quality of these procedures.

Instead of using stopwords lists, we use our NLP pipeline to to remove uninformative terms, transforming the text using lexical roots (i.e., the lemmas) to eliminate noise while reducing redundant linguistic variability. Only nouns are considered as informative features. Our approach does not make use of WordNet-like methods which only have top-level domain terms, which often fail to recognize and lemmatize the legal terminology.

[5] https://ec.europa.eu/jrc/en/publication/contributions-conferences/jrc-acquis-multilingual-aligned-parallel-corpus-20-languages

Concerning the numeric representation of the text, we use our statistical framework to compute the Term Frequency-Inverse Document Frequency (TF-IDF) term weights.

5.1.3 Multi-Label to Mono-label Transformation

As seen in the related work section, the transformation of a multi-label text corpus into mono-labels can be achieved with several strategies. In this work we used an idea that comes from an approach mentioned in [42]. The transformation of a multi-label dataset into a mono-label dataset enables the use of a standard Support Vector Machine classifier, that is known to be the best choice when dealing with textual databases.

The general idea is that one n-labeled document can be seen as a collection of n different documents. Since a document is represented by a numerical vector (according to the Vector Space Model [39]), it can also be viewed as a fusion of multiple single-labeled vectors. This, however, is based on the assumption that only one feature belongs to one label, which is clearly a distortion of the reality.

Given the vectorial representation of a text $\mathbf{d}$ and its set of associated labels S_d, the system splits the document into $|S_d|$ virtual documents, each one belonging to one label. With this technique, all the multi-label original vectors are separated in mono-label vectors that can be used in a standard SVM-based classification environment.

5.1.4 Evaluation

In flat classification scenarios, it is common practice to evaluate classification systems by means of *Precision* and *Recall* (and *F-Measure*). While Precision is the fraction of retrieved instances that are relevant, Recall is the fraction of relevant instances that are retrieved. F-Measure is the harmonic mean of Precision and Recall.

In multi-label classification contexts, accuracy is often calculated by averaging Precision, Recall, and F-Measure values. There are two conventional methods of calculating these average values: Micro-average gives equal importance to each document and thus it uses a global contingency table to compute the accuracy values. Macro-Average instead calculates Precision and Recall for each category and then takes the average of these. In our experiment we evaluated the system by using the Micro-average system, thus giving to each document the same importance.

As can be seen in Table 1, the use of SVM, rather than distance-based classification approaches like Cosine Similarity, brings higher degrees of accuracy.

5.2 Concept and Relation Extraction

In this section we present our approach to identifying semantic concepts and relations between legal texts and semantic entities. For our experiments we

Approach	#Cat	Prec.	Recall	F-Meas.
JEX ([40])	2688	47.13%	54.64%	50.61%
Our system	3820	**70.64%**	**79.70%**	**74.90%**

Table 1 Accuracy levels for the classification of the JRC-Acquis corpus (Italian version) into the first six more probable categories.

initially focus on three types of semantic labels (or tags): *active roles*, *passive roles*, and *related notions*, i.e., concepts in the text without specific relations.

Our methodology consists in seeing the problem as follows: given a set of semantic annotations *S(x)* between a syntactic chunk x and the semantic tag S, the task is to feed a SVM-classifier with their syntactic context to be able to generalize over these semantic connections. All the nouns y that are not associated with the semantic tag S are used as negative examples. This way, the classifier is asked to learn a syntactic model of the chunks that underlies the semantic annotation S. Then, when parsing new text, all its syntactic chunks are passed through the S-based classifier that decides if they can be annotated with S or not.

The problem of finding a relation between a term and a semantic label is faced by using the term's local syntactic information. Dependency parsing is a procedure that extracts syntactic dependencies among the terms contained in a sentence, such as modifiers of nouns, arguments of verbs, and so forth. The idea is that a semantic tag may be characterized by limited sets of syntactic contexts. According to this assumption, the task can be seen as a classification problem where each term in a sentence has to be associated with a specific semantic label given its syntactic dependencies.

The process starts as follows: the syntactic dependencies given by the NLP module are transformed into abstract textual representation in the form of triples. In particular, for each syntactic dependency *dep(a,b)* (or *dep(b,a)*) of a considered noun a, we create an abstract term *dep.target.B* (or *dep.B.target*, where B becomes the generic string *NOUN* in case it is a noun (as opposed to a); otherwise it is equal to b. This way, the nouns are transformed into textual abstractions. This procedure creates a level of generalization of the features that collapses the variability of the nouns involved in the syntactic dependencies.

Given a legal text T, the system produces as many input instances as the number of nouns contained in T. In particular, for each noun n in T, and for each semantic tag S, we produce an instance $T^n sem$ associated with the label positive if n has been annotated with S in the training corpus (negative, otherwise). At the end of this process, all the instances are transformed into numeric vectors according to the Vector Space Model, and they are finally used as the input training set for a Support Vector Machine classifier. This is done for all the semantic information that we tested; this means that we build three classifiers, one for each semantic tag. Once the classifiers are built, we can

classify all the nouns of a text as belonging to one of the three semantic labels (or none of them) by passing their syntactic dependencies to such classifiers.

Our approach is susceptible to errors given by the POS-tagger and the syntactic parser. In fact, where the POS-tagger does not recognize that a target term is actually a noun, then the latter is not considered as a possible item that deserves a semantic label. The same thing happens for the parsing procedure, which can produce errors that avoid the correct classification of a target noun. The approach works almost perfectly with the active role semantic tag, with a Precision of 97.2% and a Recall of 92.6%. This means that the syntactic context of the active roles is stable, so it is easy for the classifier to build the model. Regarding the passive role tag, even if the approach is precise when identifying the right semantic label (100% Precision), it returns many false negatives (26.8% Recall). In a semi-supervised context of an ontology learning process, this can in any case provide good support, since all of what has been automatically identified is likely to be correct. Finally, the involved object semantic tag gave quite low results in terms of Precision and Recall (59.3% and 31.9% respectively). On average, only six to ten nouns classified as involved objects were actually annotated with the right semantic label. This is due to the very wide semantic coverage of this specific tag, and its consequently broad syntactic context. In the future, we plan to extend the module for concept and relations mining by integrating flat reification-based representations such as the ones proposed in [33], [34], and [37].

5.3 Extracting legal modifications

A knowledge engineer would manually specify whether the reference is a simple reference or it modifies or overrides other legislation, and would input all the elements of the modification, such as the date, target, etc. We instead use our rule-based pattern matching technologies to automate the process of adding the <*inlinemeta*> tag for each modificatory clause.

The format of the pattern-matching rules used in Eunomos has been described above in Section 4.2. For ease of understanding, we provide only conceptual representations in the figures below. Figure 6 shows an example of instance for the pattern in Figure 4. The rule is triggered when the system finds in the input text a verb with the lemma 'sopprimere' (*to suppress*).

Then, it checks whether there is a verb with lemma 'essere' (*to be*) between the two[6] and their preceding words, and whether there is a normative reference among the five preceding words of the lemma 'essere'. The normative reference is a portion of text referring to a law or an article within a law. The NIR documents downloaded from Normattiva specify most normative references,

[6] We specified a maximum distance of 2 words in order to encompass both sentences of the form 'Il rif1 è soppresso' (*The rif1 is suppressed*) and sentences of the form 'Il rif1 è stato soppresso' (*The rif1 has been suppressed*). In Italian, the lemma of both words 'è' and 'stato' is 'essere'.

therefore in this process they can be substituted with the strings `rif1`, `rif2`, etc. and considered as proper nouns by the TULE parser.

When the rule in Figure 6 is satisfied, the provision is annotated as 'abrogazione', with the normative reference occurring therein identified as 'norma'.

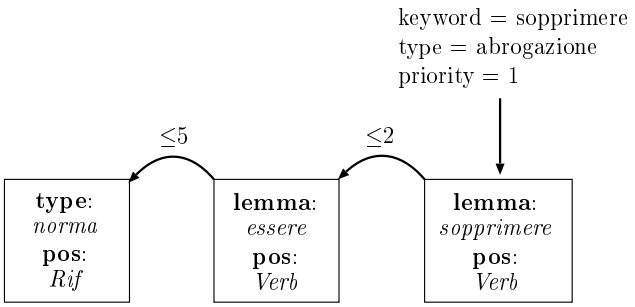

Fig. 6 A rule for some kinds of 'abrogazioni' (*abrogations*)

Many provisions are correctly classified by the rule in Figure 6. Nevertheless, the rule can also lead to wrong annotations. Although the main verb of some provisions is 'sopprimere', the text is technically a 'sostituzione'. Generally, sentences of the form 'Il rif1 è soppresso da rif2' (*The rif1 is suppressed by rif2*) are substitutions, not abrogations.
Thus, we add in the system the rule in Fig.7, and of course assign to it a higher priority than the rule in Fig6, so that it is executed before the latter.

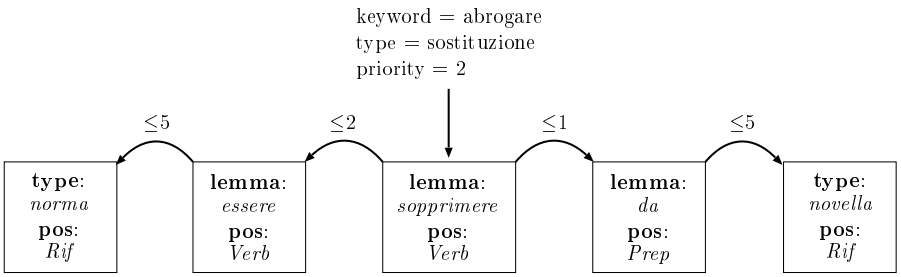

Fig. 7 A rule for certain kind of 'sostituzioni' (*substitutions*)

The checks carried out on the words preceding the keyword 'sopprimere' are the same as for those in Figure 6. Furthermore, the rule in Figure 7 requires the occurrence of the preposition 'da' immediately after the keyword and a normative reference (that will be annotated as 'novella') among the five words following the preposition.

To evaluate the module for extracting legal modifications, we used a dataset composed of 180 files, containing 2,306 modificatory provisions manually an-

notated by the legal experts of the CIRSFID research center[7] of the University of Bologna.

Our system obtains 98.56% precision and 86.60% recall. The match between a provision automatically calculated by the module and the corresponding one stored in the corpus is considered valid only if it matches both the type of the provision (abrogation, substitution, insertion, etc.) and *all* its arguments, such as "norma" and "novella" in Fig.7.

It is worth noticing that the system presented here achieves an higher level of precision, close to 100%, because the rules behave as a kind of "filter". In other words, the system uses *ad-hoc* rules, each of which describes a specific valid pattern. As a consequence, (almost) any provision matching with this pattern is precisely classified by the pattern itself. Recall is lower in that rules are added one by one, which turns out to be an highly time-consuming task. Our future developments in Eunomos include the implementation of a web interface for allowing legal experts to quickly tag the missing provisions, in order to inform the IT experts that a rule covering that particular linguistic pattern has to be added to the module for extracting legal modifications.

5.4 Entity Linking

In the proposed system, the process of linking terms in the text of the law to concepts in the ontology is carried out via the pattern-matching rule tool briefly described above. However, the linking process is semi-automatic. A (human) legal expert, via a special web interface, must *validate* the links suggested by the tool. It is often the case that the same portion of legal text could be linked to different concepts in the ontology, in particular because several legal terms are indeed substrings of other legal terms. As such, the web interface shows *all* legal terms recognized by the NLP modules, and the interface allows the legal expert to select the one that is more appropriate in that context. Once the concept is selected, the interface automatically creates the link to the concept.

In particular, two additional attributes in the rule specify the conceptId and the domainId. The rule system encloses the (contiguous) text over which the rule spans within the `concept` tag:

```
<concept id=X domain=Y>...text...</concept>
```

where `X` and `Y` are respectively the values of the two attributes conceptId and the domainId.

A concrete example of a pattern-matching rule is shown below. The rule recognizes the legal term "conflitto di interesse", and associates it with conceptId=4399 and domainId=11.
This rule is automatically generated from the Eunomos database. The legal terms are parsed via TULE, in order to recognize all content words (nouns,

[7] http://www.cirsfid.unibo.it

verbs, adjectives, and adverbs). Then, a rule is built using only content words. The rule involves the nouns “conflitto” and “interesse” and it is satisfied on new text if it contains the noun “conflitto” followed, after at most two words (`maxDistance="2"`), by the noun “interesse”.

If the rule is satisfied and the legal expert validates the result, a link from text to the conceptId=4399 in the domainId=11 is created.

```
<rule value="conflitto">
  <constraint conceptId="4399" domainId="11">
    <headAlternatives>
      <head>
        <Lemma>conflitto</Lemma>
        <Pos>Noun</Pos>
      </head>
    </headAlternatives>
    <nextAlternatives>
      <next maxDistance="2">
        <headAlternatives>
          <head>
            <Lemma>interesse</Lemma>
            <Pos>Noun</Pos>
          </head>
        </headAlternatives>
      </next>
    </nextAlternatives>
  </constraint>
</rule>
```

The Entity Linking module allowed so far to create a corpus of 500 documents among compliance, violations, penalties, etc. that have been fully annotated with respect to the concepts of the Syllabus ontology.

6 Related Work

In this section we present related techniques concerning the extraction of semantic knowledge from texts.

A first task is the identification of references/citations among sub parts of a single document [41]. To achieve text-to-ontology linking, textual documents are usually converted into XML documents that contain special tags/attributes that specify the IDs of the ontology nodes. A recent example is the LEMON (Lexicon Model for Ontologies) system. Lemon is a proposed model for modeling lexicon and machine-readable dictionaries, linked to the Semantic Web and the Linked Data cloud. It was designed to specifically separate the lexicon from the ontology layers.

The presented system has been designed in line with the same principles as the LEMON one. As described above in subsection 3.2, the Syllabus ontology separates the definition of concepts from the grammar rules used to express them in a particular language. The latter are used to guide the linking of the textual chunks occurring in the documents to the concepts in the ontology.

6.1 Ontology Learning in the Legal Domain

To the best of our knowledge, there is still very little research concerning ontology learning and semantic search in the legal domain, with respect to the open-domain literature. Most efforts have been dedicated to standard classification tasks. [15], for instance, used a set of rules to find patterns suggestive of a particular semantic class. Their classification task was quite different from ours since their classes were types of norms like delegations and penalizations, while we categorize single syntactic chunks as related to specific topic labels, so with a different level of granularity. [4] achieved an accuracy of 92% in the task of classifying 582 paragraphs from Italian laws into ten different semantic categories such as 'Prohibition Action', 'Obligation Addressee', 'Substitution', and so on. [28] proposed a method to detect modificatory provisions, i.e., fragments of text that make a change to one or more sentences in the text or in the normative arguments. [8] proposed a supervised technique to identify semantic relations through the embedding of syntactic information within an SVM classifier.

According to [5] and [11], the problem of extracting ontologies from text can be faced at different levels of granularity. Similarly to the former, our approach belongs to the extraction of *terminological ontologies* based on IS-A relations, while similarly to the latter we refer to the *concept hierarchies* of their *Ontology Learning layer cake*. As for the task of definition extraction, most of the existing approaches use symbolic methods that are based on lexico-syntactic patterns, which are manually crafted or deduced automatically. The seminal work of [23] represents the main approach based on fixed patterns such as "NP_x *is a/an* NP_y" and "NP_x *such as* NP_y", that usually imply $< x$ IS-A $y >$. The main drawback of such a technique is that it does not face the high variability of how a relation can be expressed in natural language. Still, it generally extracts single-word terms rather than well-formed and compound concepts. The work of [31][43] is based on graph structures that generalize over the POS-tagged patterns between x and y. [3] and [24] proposed similar lexico-syntactic patterns to extract *part-whole* relationships. [16] proposed a rule-based approach for the extraction of hypernyms that, however, leads to very low accuracy values in terms of Precision. [32] proposed a technique to extract hypernym relations from Wikipedia by means of methods based on the connectivity of the network and classical lexico-syntactic patterns. [44] extended their work by combining extracted Wikipedia entries with new terms contained in additional web documents, using a distributional similarity-based approach. [30] proposed a technique that uses parse subtree kernels to classify predicate-argument attachments, demonstrating the efficacy of using syntactic information rather than patterns. However, our method represents a computationally lighter approach since the feature space is limited.

Finally, pure statistical approaches present techniques for the extraction of hierarchies of terms based on word frequency as well as co-occurrence values, relying on clustering procedures [12][18][21][45][17]. The central hypothesis is that similar words tend to occur together in similar contexts [22]. Despite

this, they are defined by [5] as prototype-based ontologies rather than formal terminological ontologies, and they usually suffer from the problem of data sparsity in the case of small corpora.

6.2 Automatic Text Classification of Multi-Labeled Texts

Regarding the automatic treatment and classification of legal text we shall refer mainly to the work of [40], since they share both our goals and the data used. In [40] the authors presented the classification system named JEX, that computes a profile for each Eurovoc category based on all the documents that are associated with it within the corpus. Such a profile is constituted by a set of pairs $<word, weight>$ (i.e., a category-vector), namely terms and relative importance (or frequency). In order to classify a text document, JEX first creates a document-vector and then finds the K most similar category-vectors by means of Cosine Similarity. The latter is a measure that captures the similarity of two texts by evaluating the lexical overlapping (and proportion) between the two.

Then, [14] presented a comparison of Machine Learning techniques versus knowledge engineering in the classification of legal sentences. In detail, the authors in [14] use a set of rules to find patterns suggestive of a particular class. Finally, [4] presented a system to classify paragraphs from Italian laws into ten different categories using SVM, reaching 92% accuracy, even if their categories were high-level meta-classes such as "Substitution", and so on.

The problem of classifying text documents associated with multiple categories is currently met in several domains and applications. SVM, like others, only works with mono-label texts, thus a pre-processing of the data is needed in that sense. Even if there are adaptations of well-known algorithms for dealing with multi-label data, the most applied approach concerns the transformation of multi-label data into mono-label. Among all techniques, there are naive solutions like the random selection of one of the multiple categories for each document as well as more complex ones. [25] [10], for instance, all use a transformation method that creates one binary classifier for each category. To classify a document, it needs to be processed from all the category-classifiers, and so it may represent a prohibitive solution in the case of thousands of categories. Another approach named *power set* considers each different set of labels associated with a document to be a single label [19]. This solution, however, may lead to data sets with a large number of classes and few examples per class.

7 Conclusions

In this paper, we described the need for innovative technologies to support compliance management. We then described the Eunomos legal knowledge management system, which helps users understand the meaning of legislative

text, and the relationship between norms. Finally, we described NLP applications to semi-automate essential time-consuming and lower-skill tasks.

The main aim of the system is to make legislation more accessible. One important aspect is having effective mechanisms for document retrieval and indexing. The classification of legislation and individual articles helps filter searches to the most relevant norms. The tool for text classification in accordance with the Eurovoc thesaurus ensures that new legislation can be classified quickly. The documents are transformed into NIR legislative XML using the ITTIG parser. The creation of hyperlinks for cross-references between legal documents helps the user understand legislation in the context of previous and subsequent legislation. The ITTIG parser identifies only references, while the proposed tool for extracting modifications helps identify the type of amendment. The Legal Taxonomy Syllabus ontology framework helps users understand the meaning of laws. The ontology of terms links concepts to all the terms that express them, and links instances of such terms in the legislation to the most relevant conceptual descriptor by using the entity linking tool. The ontology of prescriptions provides a way to structure norms in a way that shows clearly all the relevant components. Finally, the concept and relation extraction tool extracts some key components from the legislative text.

The developed NLP tools are mainly focused and tailored on the Italian language. However, algorithms are language-independent and they can be easily re-implemented with the use of (often much more developed) English-based resources for language understanding.

In conclusion, maintaining accurate legal knowledge management requires continuous effort involving lower-skill tasks as well as legal expertise. Using NLP tools to semi-automate the lower-skill tasks makes this ambitious project a realistic commercial prospect as it helps keep costs down while at the same time allowing greater coverage. The presented system can be employed as an in-house software that enables expert users to search, classify, annotate and build legal knowledge and keep up to date with legislative changes. Alternatively, it can be offered as an online service so that legislation monitoring is effectively outsourced. The software and related services can be provided to several clients, which means that information and costs are shared.

References

1. Gianmaria Ajani, Guido Boella, Luigi Di Caro, Livio Robaldo, Llio Humphreys, Sabrina Praduroux, Piercarlo Rossi, and Andrea Violato. The european legal taxonomy syllabus: a multi-lingual, multi-level ontology framework to untangle the web of european legal terminology. *Applied Ontology*, 2017.
2. Gianmaria Ajani, Leonardo Lesmo, Guido Boella, Alessandro Mazzei, and Piercarlo Rossi. Terminological and ontological analysis of european directives: multilinguism in law. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law: ICAIL*, pages 43–48. ACM, 2007.
3. M. Berland and E. Charniak. Finding parts in very large corpora. In *Annual Meeting Association for Computational Linguistics*, volume 37, pages 57–64. Association for Computational Linguistics, 1999.

4. Carlo Biagioli, Enrico Francesconi, Andrea Passerini, Simonetta Montemagni, and Claudia Soria. Automatic semantics extraction in law documents. In *Proceedings of The Tenth International Conference on Artificial Intelligence and Law: ICAIL*, pages 133–140. ACM, 2005.
5. C. Biemann. Ontology learning from text: A survey of methods. In *LDV forum*, volume 20, pages 75–93, 2005.
6. Guido Boella, Luigi Di Caro, Michele Graziadei, Loredana Cupi, Carlo Emilio Salaroglio, Llio Humphreys, Hristo Konstantinov, Kornel Marko, Livio Robaldo, Claudio Ruffini, et al. Linking legal open data: breaking the accessibility and language barrier in european legislation and case law. In *Proceedings of the 15th International Conference on Artificial Intelligence and Law*, pages 171–175. ACM, 2015.
7. Guido Boella, Luigi Di Caro, Llio Humphreys, Livio Robaldo, and Leon van der Torre. Nlp challenges for eunomos, a tool to build and manage legal knowledge. *Language Resources and Evaluation (LREC)*, pages 3672–3678, 2012.
8. Guido Boella, Luigi Di Caro, and Livio Robaldo. Semantic relation extraction from legislative text using generalized syntactic dependencies and support vector machines. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, pages 218–225. Springer, 2013.
9. C. Bosco, A. Montemagni, A. Mazzei, V. Lombardo, F. Dell'Orletta, A. Lenci, L. Lesmo, G. Attardi, M. Simi, A. Lavelli, J. Hall, J. Nilsson, and J. Nivre. Comparing italian parsers on a common treebank: the evalita experience. In *Proc. of the 6th Int. Conf. on Language Resources and Evaluation (LREC 2010)*, 2010.
10. M.R. Boutell, J. Luo, X. Shen, and C.M. Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.
11. P. Buitelaar, P. Cimiano, and B. Magnini. Ontology learning from text: An overview. *Ontology learning from text: Methods, evaluation and applications*, 123:3–12, 2005.
12. K.S. Candan, L. Di Caro, and M.L. Sapino. Creating tag hierarchies for effective navigation in social media. In *Proceedings of the 2008 ACM workshop on Search in social media*, pages 75–82. ACM, 2008.
13. C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
14. E. de Maat, K. Krabben, and R. Winkels. Machine learning versus knowledge based classification of legal texts. In *Proceedings of Legal Knowledge and Information Systems Conference: JURIX 2010*, pages 87–96, 2010.
15. Emile de Maat, Kai Krabben, and Radboud Winkels. Machine learning versus knowledge based classification of legal texts. In *Proceedings of Legal Knowledge and Information Systems Conference: JURIX 2010*, pages 87–96. IOS Press, 2010.
16. R. Del Gaudio and A. Branco. Automatic extraction of definitions in portuguese: A rule-based approach. *Progress in Artificial Intelligence*, pages 659–670, 2007.
17. Luigi Di Caro, K Selçuk Candan, and Maria Luisa Sapino. Using tagflake for condensing navigable tag hierarchies from tag clouds. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1069–1072. ACM, 2008.
18. Luigi Di Caro, K Selçuk Candan, and Maria Luisa Sapino. Navigating within news collections using tag-flakes. *Journal of Visual Languages & Computing*, 22(2):120–139, 2011.
19. S. Diplaris, G. Tsoumakas, P. Mitkas, and I. Vlahavas. Protein classification with multiple algorithms. *Advances in Informatics*, pages 448–456, 2005.
20. Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
21. B. Fortuna, D. Mladenič, and M. Grobelnik. Semi-automatic construction of topic ontologies. *Semantics, Web and Mining*, pages 121–131, 2006.
22. Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954.
23. M.A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics, 1992.

24. Ashwin Ittoo and Gosse Bouma. Minimally-supervised extraction of domain-specific part-whole relations using wikipedia as knowledge-base. *Data Knowl. Eng.*, 85:57–79, 2013.
25. B. Lauser and A. Hotho. Automatic multi-label subject indexing in a multilingual environment. *Research and Advanced Technology for Digital Libraries*, pages 140–151, 2003.
26. L. Lesmo. The Rule-Based Parser of the NLP Group of the University of Torino. *Intelligenza Artificiale*, 2(4):46–47, June 2007.
27. Leonardo Lesmo. The turin university parser at evalita 2009. *Proceedings of EVALITA*, 9, 2009.
28. Leonardo Lesmo, Alessandro Mazzei, Monica Palmirani, and Daniele P Radicioni. Tulsi: an nlp system for extracting legal modificatory provisions. *Artificial Intelligence and Law*, pages 1–34, 2013.
29. G.A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
30. Alessandro Moschitti and Cosmin Adrian Bejan. A semantic kernel for predicate argument classification. In *CoNLL-2004*, 2004.
31. Roberto Navigli and Paola Velardi. Learning word-class lattices for definition and hypernym extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1318–1327, Uppsala, Sweden, July 2010. Association for Computational Linguistics.
32. S.P. Ponzetto and M. Strube. Deriving a large scale taxonomy from wikipedia. In *Proceedings of the national conference on artificial intelligence*, volume 22, page 1440. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.
33. L. Robaldo. Interpretation and inference with maximal referential terms. *The Journal of Computer and System Sciences*, 76(5):373–388, 2010.
34. L. Robaldo. Distributivity, collectivity, and cumulativity in terms of (in)dependence and maximality. *The Journal of Logic, Language, and Information*, 20(2):233–271, 2011.
35. L. Robaldo, T. Caselli, I. Russo, and M. Grella. From italian text to timeml document via dependency parsing. In *proc of the 12th International Computational Linguistics and Intelligent Text Processing Conference (CICLing 2011), Tokyo, Japan, 2011.*, pages 177–187, 2011.
36. L. Robaldo, L. Di Caro, and A. Antonini. Sentitagger - automatically tagging text in opinionmining-ml. In *ESSEM@AI*IA*, volume 1096 of *CEUR Workshop Proceedings*, pages 177–180. CEUR-WS.org, 2013.
37. L. Robaldo and X. Sun. Reified input/output logic: Combining input/output logic and reification to represent norms coming from existing legislation. *The Journal of Logic and Computation*, 7, 2017.
38. G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.
39. G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18:613–620, November 1975.
40. R. Steinberger, E. Mohamed, and M. Turchi. Jrc eurovoc indexer jex-a freely available multilabel categorisation tool. In *In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC2012)*, 2012.
41. Oanh Thi Tran, Ngo Xuan Bach, Minh Le Nguyen, and Akira Shimazu. Automated reference resolution in legal texts. *Artif. Intell. Law*, 22(1):29–60, 2014.
42. G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.
43. Paola Velardi, Stefano Faralli, and Roberto Navigli. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. 2012.
44. I. Yamada, K. Torisawa, J. Kazama, K. Kuroda, M. Murata, S. De Saeger, F. Bond, and A. Sumida. Hypernym discovery based on distributional similarity and hierarchical structures. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 929–937. Association for Computational Linguistics, 2009.
45. H. Yang and J. Callan. Ontology generation for large email collections. In *Proceedings of the 2008 international conference on Digital government research*, pages 254–261. Digital Government Society of North America, 2008.