Automated Rules Generation from Natural Language Legal Texts

#466

Abstract

Legal texts express conditions in natural language describing what is permitted or forbidden or mandatory in the context they regulate. Normative reasoning techniques allow to reason over the set of permitted/forbidden/obligatory actions in order to detect possible violations or to check whether the behavior of the involved actors is compliant or not with respect to such actions. However, the main problem in this pipeline is that of moving from a normative text in natural language to a set of rules of the kind if A occurred, then B is obligatory expressed in a (semi)formal way. In this paper, we propose and evaluate a natural language processing framework that allows us to automatically move from a natural language legal text to a set of rules extracted from such text. The framework is discussed and evaluated exploiting the Australian "Telecommunications consumer protections code".

1 Introduction

Applying normative reasoning techniques to real world scenarios has to face the challenge of processing natural language texts. On the one side, all codes and normative documents of public institutions and companies are expressed in natural language, and it is very rare to have some kind of structured representation of the norms contained in these documents. On the other side, normative reasoning techniques need some kind of formalization of such norms in order to infer further information or to check whether the behavior observed in the involved actors is compliant with the norms, or whether a violation occurs. In this kind of formal frameworks, the standard basic representation of a norm is under the form of a set of rules that express what is permitted, forbidden or obligatory. An example of normative rule is as follows: $received_complaint \Rightarrow [Obl]inform_consumer_process$ meaning that a supplier has to inform the consumer of the complaint procedure upon reception of a complaint. This kind of rules are not always identifiable in a clear way in a normative text that contains both rules and general purpose text. This task, usually addressed by humans, is difficult even for them, and it becomes challenging for an automated system. Furthermore,

this task is not only difficult but very time consuming for humans, and (even partially) automating such a task to reduce the amount of work demanded to humans would be valuable.

The research question we answer in this paper is: *How to extract in an automated way a set of rules from natural language legal texts?* This question breaks down into the following subquestions: *1)* How to extract and distinguish the *terms* to be part of the rules?, *2)* How to identify the deontic component of each rule?, and finally *3)* How to construct the rules with the correct assignment of the terms to the antecedent and consequent of the rules?.

To answer these questions, we adopt Natural Language Processing (NLP) techniques. More precisely, our framework for automated rules generation exploits the Stanford Parser to obtain the grammatical representation of the sentences. From this structure, the relevant terms are identified (i.e., those terms to be combined in the rules), exploiting in addition a lightweight ontology we constructed to represent how the text is structured in the specific normative document we analyze. For instance, in the example above the two terms are received_complaint and inform_consumer_process. As a second step, exploiting a lightweight ontology that defines some basic formulation of permissions, prohibitions and obligations in legal texts, the deontic component to be associated to each term is identified, if any. Note that not all terms are associated to a deontic component, e.g., in the rule above term received_complaint is not associated the deontic component while inform_consumer_process does. Finally, rules are constructed by combining the identified terms, and the deontic component. Each step of the task is evaluated on a section of the "Telecommunications consumer protections code".

The advantage of our approach is that there is no need to learn how to extract the rules building a huge annotated data set of legal documents. The results are encouraging and show that our framework can actually be exploited to support humans in the rules identification task, providing considerable gains from the time and the value perspective.

The remainder of this paper is as follows: Section 2 discusses the related literature and compares it with the proposed approach, Section 3 describes the two lightweight ontologies we constructed to support the identification of the terms and of the deontic component, and presents the overall framework for automated rules generation. Section 4 presents the evaluation setting, and finally conclusions are drawn.

2 Related Work

Several ontologies have been designed to model legal concepts. Among others, there are the Functional Ontology for Law (LFU) [Valente and Breuker, 1994] about normative knowledge, world knowledge, and responsibility knowledge, the Frame-Based Ontology of Law (FBO) [van Kralingen, 1997] about norms, acts and concepts descriptions, the IKF-IF-LEX Ontology for Norm Comparison [Gangemi and Breuker, 2002] about agents, institutive norms, instrumental provisions, regulative norms and norm dynamics, the LKIF-Core Ontology ¹ including the OWL ontology of fundamental legal concepts [Rubino et al., 2006]. Moreover, [Boella et al., 2004] have proposed an ontological model of norms based on the concept of agency, and they have addressed the issue of representing interpretation of terms besides the definitions occurring in the EU directives [Ajani et al., 2010]. Finally, [Gangemi et al., 2003] showed how legal ontologies can be exploited to create newly designed legal decision support systems, and he proposed design patterns for legal ontologies [Gangemi, 2009]. However, all these works go beyond (and more far) the aim of our lightweight ontology of legal lexicon that has the basic goal of associating natural language expressions occurring in legal texts to the three deontic modalities (i.e., permissions, prohibitions and obligations) used in deontic rules.

Concerning the automated processing of legal texts to extract some kind of information, there are a number of approaches addressing this problem. [Soria et al., 2005] address an automated processing of legal texts exploiting NLP techniques but with a different goal with respect to the present paper: they aim at classifying law paragraphs according to their regulatory content, while our goal is to extract in an automated way rules with deontic modalities from legal texts. [Biagioli et al., 2005], instead, propose an automated framework for the semantic annotation of provisions to ease the retrieval process of norms, [de Araujo et al., 2013] present a knowledge extraction framework from legal texts, and [Kiyavitskaya et al., 2008] present a tool for extracting requirements from regulations where texts are annotated to identify fragments describing normative concepts and then a semantic model is constructed from these annotations and transformed into a set of requirements. Also in these cases, the goal of the automated processing of legal texts is different. [Emani, 2014] analyzes a problem close to our one, such as how to extract business rules in an automated way from regulatory texts, to verify their consistency. The paper does not propose an actual implemented solution nor an evaluation, but it discusses many open problems in addressing this task highlighting the high degree of complexity of the goal we share. [van Engers et al., 2004] present an automated concept and norm extraction framework that adopts linguistic techniques. The goal of this paper is the same as ours: an automated norm/rules extraction system will help in saving knowledge analysts a lot of time and it also contributes to a more uniform knowledge representation of such formal norms/rules. However, the adopted methodology is different: they exploit Juridical (Natural) Language Constructs (JLC) that formalize legal knowledge using NLP by introducing a set of predefined natural language constructs to

define a subset of all possible legal sentences. This kind of "patterns" is identified in the text thanks to the identification of noun and verb phrases, and then they are translated into formal rules. We do not need to define such a kind of construct, as we rely directly on the structured representation of the sentence returned from the parser, and on the three steps performed by our system namely the extraction of terms, modalities, and finally rules. Moreover, they do not consider the identification of deontic modalities in rules, and no evaluation of the automated norms extraction framework is provided. [de Maat and Winkels, 2010] use machine learning for Dutch regulations, [Brighi and Palmirani, 2009] and [Francesconi, 2010] do the same for Italian ones. These approaches classify documents or sentences, differently from our methodology where rules are extracted from the structural representation of legal texts. Finally, [Wyner and Peters, 2011] present a linguistics based approach to extract deontic rules from regulations. As underlined by the authors, Stanford parser has not been evaluated against legal sources, that is the what we do in our own framework and they do as well. However, we do not exploit the General Architecture for Text Engineering, our approach does not require to annotate the legal texts, and we support the result of the parser providing our framework with the structure of the text to be analyzed. An experimental comparison with the performances reported in these works is difficult as the data sets used to evaluate them are not available nor the systems.

3 An NLP Approach to Automated Rules Generation

The NLP approach implemented in the proposed work uses several components in order to automatically generate rules from natural language text. In particular, it exploits the following elements described in more details later in this section:

- a lightweight ontology describing the deontic linguistic elements allowing the identification of the obligations, permissions, and prohibitions in legal texts;
- a lightweight ontology describing how the natural language text is represented using a certain structure and how punctuation can be interpreted for helping the extraction of rules²;
- a NLP library, namely, the Stanford Parser library³, used for parsing natural language sentences in order to have their grammatical representation. Such a representation is used from the system for the automatic extraction of the rules.

Figure 1 shows the pipeline describing how the proposed approach works. Each task is detailed in the remainder of the section.

¹http://www.estrellaproject.org/lkif-core/

²Note that these ontologies are explicitly called *lightweight* ontologies as they are not expected to be used to normalize the concepts of legal text by mapping the legal terms into concepts in ontology and obtain the meaning of the text by using the ontology structure. They are provide support for detecting the deontic components in legal texts and the structure of such texts, respectively.

³http://nlp.stanford.edu/software/lex-parser.shtml

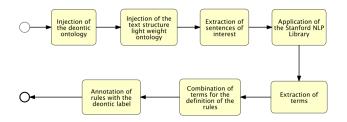


Figure 1: The pipeline of the proposed approach for rule extraction from text.

3.1 Deontic Lightweight Ontology

The deontic lightweight ontology, called normonto, has been designed in order to support the system in the automated identification of the normative component of the rules. More precisely, this ontology is exploited to identify whether a term expresses a prohibition, a permission, or an obligation. Even if several ontologies have been proposed in the latest years to represent such a kind of knowledge in different contexts, like for instance the Open Digital Rights Language (ODRL) Ontology⁴ or the LKIF-Core Ontology of basic legal concepts⁵, the aim of the normonto ontology is not to represent and model legal concepts but to specify the lexicon used to express permissions, prohibitions and obligations in natural language legal texts. For this reason, as visualized in Figure 2a, we specify the three main concepts called Obligation, Permission, and Prohibition, and the general class LexicalTerm that is further detailed into lexical terms specifying obligations (LexicalTermObl), permissions (LexicalTermPer), and prohibitions (LexicalTermPro). The lexicon used to express the normative component in legal texts is represented in the ontology as individuals of such subclasses (Figure 2b). For instance, the individual must identifies an obligation, thus it belongs to the class LexicalTermObl, and the individual not_be_allowed identifies a prohibition, thus belonging to the class LexicalTermPro. The ontology specifies that an Obligation, a Permission, and a Prohibition are expressed in the texts through specific LexicalTerm concepts, namely LexicalTermObl, LexicalTermPer, LexicalTermPro. Note that this ontology is intended to be general purpose and extensible, and differently from the text structure ontology we present in the next section, it can be exploited by the system to extract the deontic component of the rules from heterogeneous legal texts. Finally, the ontology is intended to model the legal lexicon in English. Further extensions to cover multilingual rules extraction are considered for future research.

3.2 Text Structure Lightweight Ontology

In order to support the NLP algorithm in the analysis of different textual structures, a lightweight ontology, defining the main elements of the text organization, has been modeled in order to effectively address our particular use case (explained in Section 4). Trivially, depending on the text structure that



Figure 2: The two components of the deontic ontology: the list of concepts (a) and part of the lexical terms defined as individuals (b).

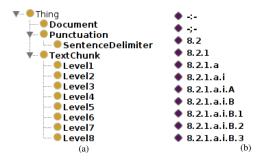


Figure 3: The two components of the text structure lightweight ontology: the list of concepts (a) and part of the lexical terms defined as individuals (b).

has to be analyzed, it might be necessary to model different lightweight ontologies dedicated to those particular purposes.

Figure 3 shows an excerpt of the modeled lightweight ontology. Concerning the concepts definition (Figure 3a), we modeled three main concepts: (i) Document, defining the conceptualization of the entire text to analyze; (ii) TextChunk, defining a single piece of text containing valuable information (i.e. antecedent or consequent of the rule that has to be extracted); and (iii) Punctuation, defining the meaning that specific punctuation signs may have in the text from the computational point of view (for instance, the ";" sign may be used for splitting sentences).

Instead, concerning individuals (Figure 3b), we modeled each block of the text as a new individual instantiating the TextChunk object. This way, we are able to represent each sentence of the text, or part of it, as a new element of the ontology in order to allow the definition of their semantic relations used by the system for the extraction of the rules.

Besides concepts and individuals, we defined two object properties (hasGeneralChunk and hasPart, the second one modeled as inverseOf of the first one) and one data property (hasText). The two object properties are used for modeling the hierarchical relationships between different TextChunk-objects; while, the hasText data property allows to associate the natural language text with the correspondent individual.

For example, we modeled the individual 8.2 as an instance of the Level1 concept, while the individual 8.2.1 as an instance of the Level2 one. Such concepts was then put in relationship by using the hasGeneralChunk object property in order to define their subsumption.

⁴http://www.w3.org/ns/odrl/2/

⁵http://www.estrellaproject.org/lkif-core/

3.3 Extraction of Sentences

The analysis of the text starts with the extraction of sentences of interest that are subsequently used for the text analysis. The extraction of such sentences is done by exploiting the structured nature of the text that generally characterize legal documents where bullet-based representation is used for describing norms contained in them. As first step, we mapped single text chunks contained in the bullet representation of the document to the lightweight ontology shown in Figure 3a. This way, we are able to manipulate a linked structure of the text easing the extraction of the full sentences. By considering the structured representation of the text as a tree, we reconstruct the set of full sentences to analyze by starting from the root of the tree and by concatenating, for each possible path, the text chunks found until the leaves are reached.

Below, we present an excerpt of the document used as test case (explained in Section 4) showing the structured representation of one of the norms contained in the document.

```
    - Acknowledging a Complaint:
    --- immediately where the Complaint is made in person or by telephone;
    --- within 2 Working Days of receipt where the Complaint is made by:
    ---- email;
    ---- being logged via the Supplier's website or another website endorsed by the Supplier for that purpose;
    ---- post; and
    ---- telephone and a message is recorded without direct contact with a staff member of the Supplier.
```

By performing the mapping between the text and the lightweight ontology, the resulting assignments are the "Level 1" to the first chunk, "Level 2" to the second and third ones, and "Level 3" to the others. By navigating through the tree representation, the sentences extracted from the text are the concatenations of the following text chunks (based on the ids written at left of each chunk): "1-2", "1-3-4", "1-3-5", "1-3-6", "1-3-7". According to what as been introduced in the previous subsection, the punctuation elements are used as regulators for deciding where to split sentences in case of complex structures. Sentences extracted at this step are then used for the extraction of the single terms as described later in this section.

3.4 The Use of the Stanford NLP Library

The extraction of rules from natural language text requires the use of tools able to provide a grammatical structure of the text that may be exploited for inferring the different components of a logical rule. The facilities available for having an effective representation of sentences are very limited. By analyzing the state of the art, one of the most prominent library is the one provided by Stanford. Such a library includes a Tree Parser able to produce a tree-based representation of each sentence and to tag them with grammatical prefixes. Moreover, the parser includes also a facility able to produce a set of grammatical relations explained which dependency elapses between two terms. The role of the parser is to work out the grammatical structure of sentences, for instance, which groups of words go together and which words are the "subject" or the

"object" of a verb. The Stanford Tree Parser is a probabilistic parser using knowledge of language gained from hand-parsed sentences to try to produce the most likely analysis of new sentences. Even if statistical parsers still make some mistakes in exceptional cases, they commonly work very well and, currently, they are the most suitable solution for a preliminary text analysis.

In the proposed approach, we decided to use the Stanford NLP library for parsing the extracted sentences and to use the produced output as starting point for terms extraction. In concrete, let us consider as example, the following sentence:

"Suppliers must demonstrate, fairness and courtesy, objectivity, and efficiency by Acknowledging a Complaint within 2 Working Days of receipt where the Complaint is made by email."

By applying the parser to such a sentence, we are able to obtain the grammatical tree of the sentence shown below⁶:

```
(ROOT
 (S
    (NP (NNS Suppliers))
    (VP (MD must)
      (VP (VB demonstrate) (, ,) (VB fairness)
        (CC and)
          (NP (NN courtesy) (, ,) (NN objectivity) (, ,)
            (CC and)
            (NN efficiency))
          (PP (IN by)
            (S
              (VP (VBG Acknowledging)
                (NP (DT a) (NN Complaint))
                (PP (IN within)
                  (NP
                     (NP (CD 2) (JJ Working) (NNS Days))
                     (PP (IN of)
                      (NP (NN receipt)))))
                (SBAR.
                  (WHADVP (WRB where))
                     (NP (DT the) (NNP Complaint))
                     (VP (VBZ is)
                      (VP (VBN made)
                         (PP (IN by)
                           (NP (NN email)))))))))))))
    (..)))
```

Such an output is used for the extraction of significant terms as described in more detail in the next section.

3.5 Extraction of Terms

Given the parsed version of each sentence, the next step is to extract relevant terms from them. With "term" we do not mean a single word (or compound names) having a meaning in a vocabulary, but we mean a complex textual expression representing an entire concept.

The extraction of the terms follows the identification of the subordinate sentences identified by the parser. In general, we

⁶For more details about the meaning of dependency used by and clause the parser, tag please refer to the official Stanford documentation: http://nlp.stanford.edu/software/dependencies_manual.pdf

interpret the beginning of a new sentence (or a subordinate one) as the beginning of a new term with some exceptions based on the content of the generated tree. Just to mention a couple of examples:

- if an extracted term starts with the expression "to VERB", the term is automatically concatenated with the previous one:
- if an extracted term contains only one token, such a token is directly concatenated to the succeeding one. This mainly happens when tokens like "where", "what", etc. are parsed.

For instance, given the sample sentence considered in the previous subsection, the analysis of the parsed representation led to the identification of the following terms:

- -: Suppliers must demonstrate fairness, and courtesy, objectivity and efficiency, by
- a: Aknowledging a Complaint within 2 Working Days of receipt
- b: where the Complaint is made by email

where, as it is possible to see, the first row is not marked as actual term but as "implicit" term. Indeed, as it will be explained in Section 4 concerning the document used as test case, some text chunks occur in many sentences. Such terms, independently by their, eventual, deontic meaning, are marked only once; while, for the other sentences, they are considered as "implicit" terms and they are not marked. The role of the "implicit" terms is to appear as antecedent of rules when, in a sentence, no terms are detected as antecedent, but consequent are identified. In the proposed example, two terms are identified.

3.6 Annotation of Terms With Deontic Tags

After the extraction of terms, they have to be annotated with the deontic tags of Obligation, Permission, and Prohibition defined in the vocabulary partially shown in Figure 2b. In the approach described in this paper, the assignment of the deontic tags is done by applying a simple text processing approach. For each extracted term, we first verify if one of the lemmatized version of the labels of the vocabulary is present in the sentence; if yes, the term is annotated with the corresponding tag. A further check is performed to verify if, for example in case of verb, the label and the "not" auxiliary have been split during the term extraction in two consecutive terms. Indeed, if this happens, the identified deontic tag has to be changed. For instance, for the labels "must" and "must not" the deontic tags used are, respectively, the "Obligation" and the "Prohibition" ones.

By taking into account the same example, the only term in which a deontic element is identified is the implicit one that is annotated with the "Obligation" tag due to the presence of the label "must".

- -: Suppliers must demonstrate fairness, and courtesy, objectivity and efficiency, by [0]
- a: Aknowledging a Complaint within 2 Working Days of receipt
- b: where the Complaint is made by email

While, for the other terms, no deontic elements have been identified, therefore, no tags have been used.

3.7 Combination of Terms For Rule Definition

The last step consists in the definition of the rules obtained by combining the extracted and annotated terms. For creating the rules, we applied a set of patterns to the terms in order to detect which are the antecedent and the consequent of each rule. Due to space reason, we are not able to report all patterns defined in the system, but only some of them:

```
[0] Term1
WHERE Term2 Rule: Term2 => [0] Term1

IF Term1
[0] THEN Term2 Rule: Term1 => [0] Term2

[0] Term1
UNLESS Term2 Rule: Term2 => [P] NOT Term1

[0] Term1
WHEN Term2
AFTER Term3 Rule: Term2 AND Term3 => [0] Term1
```

It is important to highlight that, in case a deontic tag is used for annotating an implicit term, such a tag is inherited by the first term following the implicit one. This happens because implicit terms are not taken into account for generating the rules.

Finally, by considering the annotated terms shown in the previous section and by applying the first pattern due to the presence of the "where" label, the generated rule is:

$$b => [0] a$$

4 Evaluation

The evaluation is based on the novel Australian Telecommunications Consumer Protections Code, TC628-2012 (TCPC) effective from September 1st, 2012, in particular Sections 8.2.1(a)–8.2.1(c) pertaining Compliant Management ⁷. The section describes the obligations a telecommunication service provider has to comply with when they receive a complaint from customer or consumer (for the purpose of TCPC, Section 2 Terms and Definitions customer or consumer are treated as synonymous).

The text under analysis contains a single top level clause (8.2.1) which is then divided in 3 subclauses. Furthermore, it contains 19 clauses at level 3, 16 clauses at level 4, and 4 level 5 clauses/conditions. The structure of the document (i.e., the organization of the clauses and their subclauses) indicates that the section contains 35 prima facie clauses.

For the evaluation we considered the rule-set manually generated for the case study reported in [Governatori and Shek, 2013].

For example, Section 8.2.1.a.(vii) contains the following text:

advising the Consumer or former Customer of the proposed Resolution of their Complaint within 15 Working Days from the date the Complaint is received in accordance with clause 8.2.1 (a);

⁷The package for replicating the generation of the files and all material can be downloaded at https://dl.dropboxusercontent.com/u/28364058/rule_extractor_system.zip

is mapped to the following prescriptive rule:

complaint => [0] inform_proposed_resolution_15_days prima facie clauses.

For the evaluation, we manually compared the rule-set manually generated by an analyst and the set of rules automatically extracted using the methodology described in the previous sections.

For measuring the effectiveness of the system, we evaluated the following outputs and we compared with respect to the ones contained in the manual created gold standard:

- the number of correct sentences extracted from the text;
- the number of correct terms identified in the extracted sentences;
- the number of correct deontic annotations performed on the identified terms;
- and the number of correct rules generated from the annotated terms.

The extraction of the sentences has been the first performed task, the number of extracted sentences was 28 out of the same number of sentences contained in the gold standard. Therefore, concerning the first output the precision and recall of the system were 100%.

The second task was the identification of the terms within sentences. The gold standard contains 65 terms extracted by the analysts; our system was able to extract 59 terms whose 49 were correct. Therefore, the recall obtained is 90.78% and the precision 83.05%, with a F-Measure of 86.74%.

Concerning the assignment of the deontic tag, 47 out of the 49 correct terms have been annotated correctly by obtaining a precision of 95.92%.

The last step was determining which of the 36 rules contained in the gold standard have a counterpart in the automatically generated rule set. A rule r in the automatically generated set has a counterpart if there is a rule s in the manually generated set such that the proposition in the right hand side (or consequent) of s is mapped to the consequent of r. The number of rules satisfying this condition is 33 out of 36. Finally, the last operation is to determine which extracted rules have a full correspondence with the manually generated rules; 24 of the automatically extracted rules have a corresponding rule in the manually generated set. This means that, as final values, we had a recall of 91.67% and a precision of 66.67%.

As we have discussed above the text we analyzed contains 35 prima facie clauses, and some of these rules to fully capture the nuances of the conditions under which some obligations holds; for example, as we have seen, Section 8.2.1.a.(xiii) splits in two clauses each requiring two rules. Furthermore, we would like to point out that the number of rules required to capture a norms could depend on the logic formalism used to reason with the rule. For example, if a condition of activation of an obligation is disjunctive, it was represented by two rules in the manually generated rule set. However, the disjunction could be represented by a single proposition encoding it in its meaning. Thus, the number of rules required to model a normative clause could depend on the underlying logic. This means that we can take as reference for recall not the actual number of rules in the reference but the number of prima-facie

clauses. In this case the rules extracted cover 31 of the 35 prima facie clauses.

Finally, we would like to point out that the examples given in this section (Section 8.2.1.a(vii) and 8.2.1a(xiii) were identified correctly. In most case incorrect rule were extracted because the propositions were identified incorrectly, or the rules required implicit terms in the left-hand side derived from the right hand side.

5 Concluding remarks

In this paper, we have presented a framework for the automated extraction of rules from legal texts. Our framework is based on two lightweight ontologies, namely the "general purpose" ontology indicating the lexicon used to express prohibitions, permissions and obligations in legal texts, and the "text specific" ontology used to express the structure of the precise legal text under consideration. Our framework exploits NLP to extract the rules, i.e., starting from the result of the parser, first the terms that will constitute in the rules are identified distinguishing them from those part of the text that do not convey relevant information from the normative point of view; second, the deontic component to be associated to the terms (if any) is identified, and finally, the antecedent and the consequent of the rules including the deontic component are determined and collapsed into a rule. The system is evaluated on the extraction of the rules from a section of the Australian TCPC, and the results foster further research in this direction.

Several steps need to be addressed as future research to improve the performances of the system. First of all, we need to capture the co-reference links that are present in legal texts. For instance, consider a section of the code that starts with Suppliers must provide Consumers with a Complaint handling process that [...]. in another part of the section, we have the following text A Supplier must take the following actions to enable this outcome [...]. How to recognize what is "this outcome"? We need to establish that a co-reference occurred such that the outcome is to provide consumers with a compliant handling that satisfies the certain requirements. Second, we need to align the terms used in the legal text with the terms we want to use in the rules. As shown in our evaluation, the difference between the hand-written rules and the automated extracted ones is that different terms are used to constitute the same rules. Moreover, we will integrate in the system knowledge bases to support the NLP module in order to provide it with further knowledge, in addition to the knowledge present in the text to be processed. Finally, we plan to apply our approach to different kind of legal text in order to improve the robustness of the framework with respect to the variability in the used legal language and the structure of the legal text.

References

- [Ajani et al., 2010] Gianmaria Ajani, Guido Boella, Leonardo Lesmo, Marco Martin, Alessandro Mazzei, Daniele P. Radicioni, and Piercarlo Rossi. Multilevel legal ontologies. In Enrico Francesconi, Simonetta Montemagni, Wim Peters, and Daniela Tiscornia, editors, Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language, volume 6036 of Lecture Notes in Computer Science, pages 136–154. Springer, 2010.
- [Biagioli *et al.*, 2005] Carlo Biagioli, Enrico Francesconi, Andrea Passerini, Simonetta Montemagni, and Claudia Soria. Automatic semantics extraction in law documents. In *ICAIL*, pages 133–140, 2005.
- [Boella *et al.*, 2004] Guido Boella, Leonardo Lesmo, and Rossana Damiano. On the ontological status of plans and norms. pages 317–357, 2004.
- [Brighi and Palmirani, 2009] Raffaella Brighi and Monica Palmirani. Legal text analysis of the modification provisions: a pattern oriented approach. In *ICAIL*, pages 238–239, 2009.
- [de Araujo et al., 2013] Denis A. de Araujo, Sandro Rigo, Carolina Mller, and Rove Luiza de Oliveira Chishman. Automatic information extraction from texts with inference and linguistic knowledge acquisition rules. In Web Intelligence/IAT Workshops, pages 151–154, 2013.
- [de Maat and Winkels, 2010] E. de Maat and R. Winkels. Suggesting model fragments for sentences in dutch laws. In *Proc. of LOAIT*, pages 19–28, 2010.
- [Emani, 2014] Cheikh Kacfah Emani. Automatic detection and semantic formalisation of business rules. In *ESWC*, pages 834–844, 2014.
- [Francesconi, 2010] E. Francesconi. Legal rules learning based on a semantic model for legislation. In *Proc. of SPLeT Workshop*, 2010.
- [Gangemi and Breuker, 2002] A. Gangemi and J. Breuker. Harmonizing legal ontologies. In *Deliverable 3,4 IST Project-2000-29243*. *Ontoweb*, 2002.
- [Gangemi *et al.*, 2003] Aldo Gangemi, Maria-Teresa Sagri, and Daniela Tiscornia. A constructive framework for legal ontologies. In *Law and the Semantic Web*, pages 97–124, 2003.
- [Gangemi, 2009] Aldo Gangemi. Introducing pattern-based design for legal ontologies. In *Law, Ontologies and the Semantic Web*, pages 53–71, 2009.
- [Governatori and Shek, 2013] Guido Governatori and Sidney Shek. Regorous: a business process compliance checker. In *ICAIL*, pages 245–246, 2013.
- [Kiyavitskaya *et al.*, 2008] Nadzeya Kiyavitskaya, Nicola Zeni, Travis D. Breaux, Annie I. Antn, James R. Cordy, Luisa Mich, and John Mylopoulos. Automating the extraction of rights and obligations for regulatory compliance. In *ER*, pages 154–168, 2008.
- [Rubino *et al.*, 2006] Rossella Rubino, Antonino Rotolo, and Giovanni Sartor. An owl ontology of fundamental legal concepts. In *JURIX*, pages 101–110, 2006.

- [Soria et al., 2005] C. Soria, R. Bartolini, A. Lenci, S. Montemagni, and V. Pirrelli. Automatic extraction of semantics in law documents. In European Press Academic Publishing, 2005.
- [Valente and Breuker, 1994] A. Valente and J. Breuker. A functional ontology of law. In *Artif. Intelligence and Law*, pages 7:341–361, 1994.
- [van Engers et al., 2004] T. van Engers, R. van Gog, and K. Sayah. A case study on automated norm extraction. In Proc. of JURIX, pages 49–58, 2004.
- [van Kralingen, 1997] R. van Kralingen. A conceptual framebased ontology for the law. In *Proc. of the 1st International Workshop on Legal Ontologies*, pages 6–17, 1997.
- [Wyner and Peters, 2011] Adam Wyner and Wim Peters. On rule extraction from regulations. In *JURIX*, pages 113–122, 2011.