Combining NLP Approaches for Rule Extraction from Legal Documents

Mauro Dragoni[1], Serena Villata[2], Williams Rizzi[3], and Guido Governatori[4]

[1] Fondazione Bruno Kessler, Trento, Italy
[2] CNRS, I3S Laboratory, France
[3] Universitá degli Studi di Trento, Italy
[4] NICTA Queensland, Brisbane, Australia
dragoni|wrizzi@fbk.eu villata@i3s.unice.fr guido@governatori.net

Abstract. Legal texts express conditions in natural language describing what is permitted, forbidden or mandatory in the context they regulate. Despite the numerous approaches tackling the problem of moving from a natural language legal text to the respective set of machine-readable conditions, results are still unsatisfiable and it remains a major open challenge. In this paper, we propose a preliminary approach which combines different Natural Language Processing techniques towards the extraction of rules from legal documents. More precisely, we combine the linguistic information provided by WordNet together with a syntax-based extraction of rules from legal texts, and a logic-based extraction of dependencies between chunks of such texts. Such a combined approach leads to a powerful solution towards the extraction of machine-readable rules from legal documents. We evaluate the proposed approach over the Australian "Telecommunications consumer protections code".

1 Introduction

Applying deontic reasoning techniques to real world scenarios has to face the challenge of processing natural language texts. On the one side, all codes and legal documents of public institutions and companies are expressed in natural language, and it is very unlikely to have a structured (possibly machine-processable) representation of the deontic conditions contained in such documents. On the other side, automated reasoning techniques need to process formal conditions to infer further information, or to check whether the observed behavior is compliant with such conditions, or whether a violation occurred. In this kind of frameworks, the basic representation of a legal rule is as follows: $sup\_received\_complaint \Rightarrow [Obl]inform\_consumer\_process$ meaning that a supplier has to inform the consumer of the complaint procedure upon reception of a complaint. Note that this kind of rules are not always clearly identifiable in legal texts, and this task is difficult even for humans, becoming challenging for an automated system. Defining systems able to tackle this task in an automated way is a main challenge that received a lot of attention in the past years from the legal information systems community, and heterogeneous approaches have been proposed, e.g., [1, 2]. This interest is due, not only to the difficulty for humans to address such a task, but also to the fact that the task is

extremely time consuming for humans, and (even partially) automating it to reduce the amount of work demanded to humans would become a valuable support.

Despite the huge number of proposed approaches, the problem of extracting rules or conditions from legal texts is still open. In this paper, we start from the observation that, given the difficulty of the task, the adoption of a single Natural Language Processing (NLP) approach to solve it would not lead to satisfiable results, as witnessed by very limited adoption of the current frameworks. The research question we answer in this paper is: *How to combine different NLP approaches 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 deal with the variability of natural language texts for the identification of the deontic components of each rule?, and *2)* How to combine a syntax-based approach and a semantic-based one to identify the terms composing each rule, and correctly assign them as being the antecedent/consequent of the rule?

To answer these questions, we adopt and combine a set of NLP techniques. More precisely, our framework for automated rules generation exploits the Stanford Parser to obtain the grammatical representation of the sentences, and WordNet[5] to deal with the variability of the language in expressing the deontic components in natural language legal texts. We combine this syntactic-based rules extraction approach, relying on the well known Stanford Parser, together with a logic-based approach, exploiting the Boxer framework [3] for the extraction of logical dependencies between chunks of text. The results of the evaluation of our combined framework on a section of the Australian "Telecommunications consumer protections code" show the feasibility of the proposed approach, and foster further research in this direction. 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 as for machine learning approaches.

The remainder of this paper is as follows: Section 2 discusses the related literature and compares it to the proposed approach. Section 3 presents the overall framework for automated rules extraction, and Section 4 describes the evaluation setting.

2   Related Work

The automated processing of legal texts to extract some kind of information is a challenge that received a lot of attention in the literature. [4] address an automated processing of legal texts exploiting NLP techniques: they aim at classifying law paragraphs according to their regulatory content and extracting text fragments corresponding to specific semantic roles relevant for the regulatory content, while our goal is to extract rules with deontic modalities from legal texts. [5], instead, propose an automated framework for the semantic annotation of provisions to ease the retrieval process of norms, [6] present a knowledge extraction framework from legal texts, and [7] 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. [1] present an automated concept and

---

[5] https://wordnet.princeton.edu/

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. Similarly to them, we define "patterns" for detecting the deontic rules, but we combine two approaches to lead to better results: we rely on the structured representation of the sentence returned from the parser and its logical one returned from Boxer. Finally, [1] do not consider the identification of deontic modalities in rules, and no evaluation of their automated norms extraction framework is provided thus results cannot be compared. [8] use machine learning for Dutch regulations, [9] and [10] 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, [2] present a linguistic-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, and our approach does not require to annotate the legal texts. To obtain satisfiable results, we combine the result of the parser together with the logical dependencies between chunks of text extracted from the document through Boxer. An experimental comparison with the performances reported in these works is difficult as the data sets used to evaluate them are not available. [11] present a framework to automatically extract semantic knowledge from legislative texts. A similarity with our work is that, instead of using pattern matching methods relying on lexico-syntactic patterns, they propose to adopt syntactic dependencies between terms extracted with a syntactic parser. The idea, on which the present paper is grounded as well, is that syntactic information are more robust than pattern matching approaches when facing length and complexity of the sentences. The difference consists in the kind of information extracted, legal rules in our case, and three semantic labels, namely active role, passive role, and involved object in their work.

3   The Framework

The combined NLP approach implemented in this paper adopts several components to automatically generate rules from natural language legal texts. In particular, it exploits the following elements described in more details later on in this section: i) a lightweight ontology describing the deontic linguistic elements allowing for the identification of the obligations, permissions, and prohibitions in legal texts; ii) a lightweight ontology describing how the natural language text is structured, and how punctuation can be interpreted for helping the extraction of rules[6]; iii) a NLP library, namely, the Stanford

---

[6] Note 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

Parser library[7], used for parsing natural language sentences to retrieve their grammatical representation. We decided to adopt Stanford Parser as it is the reference parser for parsing natural language sentences in English; iv) a Combinatory Categorial Grammar (CCG) parser tool including the Boxer framework [3], used for extracting logical dependencies between chunks of text from the document.

The resulting combined framework is an extension of the approach presented in [12]. In particular, the following drawbacks have been addressed with respect to [12]: i) the deontic ontology has been extended by extracting from WordNet [13] all synsets related to the meaning of the `Obligation`, `Permission`, and `Prohibition` concepts (as described in subsection 3.1). In this way, we are able to improve the precision of the term annotation activities with the deontic labels; ii) the set of the patterns used for detecting deontic rules has been enriched; iii) a parallel branch integrating the functionalities of the CCG parser has been integrated to analyze the text from a different perspective. The analysis results obtained by the CCG parser are then merged with the output of the NLP-only branch for extracting the final set of rules. Figure 1 shows the pipeline of the proposed framework.
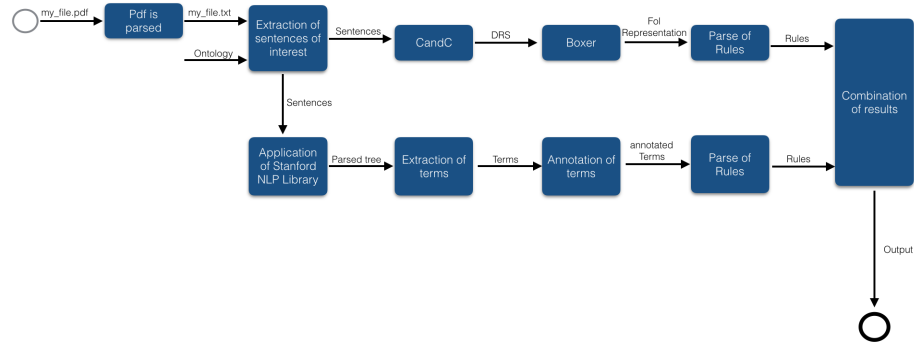

Fig. 1: The pipeline of the proposed framework.

After the preliminary steps consisting in the extraction of the text from source documents , and the composition of the separated sentences generated by the extractor, the structured representation of the text follows two parallel branches implementing two different analysis techniques. In the lower branch, the modules of the Stanford NLP library are applied for tagging sentence content, and building the related tree for extracting the terms contained in each sentence. Then, the deontic ontology is applied to annotate each term with the appropriate label, i.e., obligation, permission, prohibition.

ontology, and obtain the meaning of the text by using the ontology structure. They uniquely provide a support for detecting the deontic components in legal texts and the structure of such texts, respectively.

[7] http://nlp.stanford.edu/software/lex-parser.shtml

Finally, the system looks for patterns within the terms set of each sentence in order to compose the rules.

In the upper branch, instead, the CCG parser is applied to the full sentence to extract logic relationships between terms. Then, the output of the CCG parser is used for confirming the rules extracted through the lower branch, and for discovering new relationships between terms that have not been detected by applying the patterns adopted by the NLP parser. Each component of the pipeline is now detailed.

3.1 Deontic Lightweight Ontology

The deontic lightweight ontology, called `normonto`, has been designed 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, 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, we specify the three main concepts called `Obligation`, `Permission`, and `Prohibition`. The lexicon used to express the normative component in legal texts is represented in the ontology as individuals of such subclasses. 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`. 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. The selection of the keywords modeled as individuals in the ontology has been performed by starting from a basic set of keywords related to the concepts of prohibition, permission, and obligation. Such a set has been used for querying WordNet to extract synonyms, hypernyms, and hyponyms that are directly connected with each element of the set of keywords mentioned above. The process has been run for three times and, after each step, the content extracted for enriching the ontology has been manually validated.

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. Depending on the text structure that has to be analyzed, it might be necessary to model different lightweight ontologies dedicated to those particular purposes.

Concerning the concepts definition, 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).

Concerning individuals, 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 define 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.

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 characterizes legal documents where a bullet-based representation is used for describing normative conditions. As first step, we map single text chunks contained in the bullet representation of the document to the lightweight ontology. In 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. Let us consider an excerpt of the document used as test case (Section 4) showing the structured representation of one of the norms contained in the document:

```
(1) - Acknowledging a Complaint:
(2) --- immediately where the Complaint is made in
        person or by telephone;
(3) --- within 2 Working Days of receipt where the
        Complaint is made by:
(4) ----- email;
(5) ----- being logged via the Supplier's website
          or another website endorsed
          by the Supplier for that purpose;
(6) ----- post; and
(7) ----- 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". As in Section 3.2, 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.

3.4 The Use of the Stanford NLP Library

The extraction of rules from natural language legal texts 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. Let us consider 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 parsing this sentence, we obtain the grammatical tree of the sentence shown below[8]:

```
(ROOT
  (S
    (NP (NNS Suppliers))
    (VP (MD must)
      (VP (VB demonstrate) (, ,) (VB fairness)
        (CC and)
        (NP
          (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))
                  (S
                    (NP (DT the) (NNP Complaint))
                    (VP (VBZ is)
                      (VP (VBN made)
                        (PP (IN by)
                          (NP (NN email))))))))))))
    (. .)))
```

---

[8] For more details about the meaning of each tag and dependency clauses used by the parser, please refer to the official Stanford documentation: `http://nlp.stanford.edu/software/dependencies_manual.pdf`

3.5 Extraction of Terms

Given the parsed version of each sentence, the next step consists 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 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. Some examples are i) if an extracted term starts with the expression "to VERB", the term is automatically concatenated with the previous one, and ii) 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. Let us consider again the sample sentence of Section 3.4 where the analysis of the parsed representation leads 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
```

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. Two terms are identified here.

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 deontic lightweight vocabulary . We assign the deontic tags by applying a 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. In the 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 label "must":

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

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 apply a set of patterns to the terms in

order to detect what is the antecedent and the consequent of each rule. Due to space reasons, we are not able to report all patterns defined in the system, but only some of them:

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

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

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

[O] Term1
WHEN Term2
AFTER Term3     Rule: Term2 AND Term3 => [O] 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 Section 3.6 and by applying the first pattern due to the presence of the "where" label, the generated rule is: `b => [O] a`.

3.8 The Use of the CCG Parser

The CCG parser has been integrated for performing a logic analysis of each sentence in order to find relationships between the contained words. Indeed, semantic representations produced by the CCG parser, known as Discourse Representation Structures (DRSs), can be considered as ordinary first-order logic formulas and can be exploited to find semantic connections between each word extracted from the sentences. The aim of the integration of such a component is to support the NLP pipeline described above in detecting the relationships between the sentences from which the Stanford parser is not able to extract any information through the application of the pattern-based mechanism (Section 3.7). The exploitation of such logical relationships allows to improve the general effectiveness of the rules extraction system.

Consider the following example of the output generated by the CCG parser (Figure 2) and of the linguistic graph that we build starting from the relationships between words found by the parser (Figure 3).

Graph's connections are exploited for two purposes. First, for sentences where deontic rules have been extracted by the NLP-only pipeline, we verify if the CCG parser finds relationships between the terms involved in the rule (the effectiveness of the pipeline by considering the different scores between these rules will be discussed in Section 4). Second, for sentences where deontic rules are not detected by the NLP-only pipeline, in particular by the pattern-based mechanism, if the CCG parser identifies logical relationships between the terms contained in the sentence, a new deontic rule is created and stored in the rules set.

4   Evaluation

The evaluation is based on the novel Australian Telecommunications Consumer Protections Code, TC628-2012 (TCPC) effective from September 1st, 2012, in particular

Fig. 2

Fig. 3

Sections 8.2.1(a)–8.2.1(c) pertaining Compliant Management. 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 example, Section 8.2.1.a.(vii) states that:

> 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);

is mapped to the following prescriptive rule:

```
complaint => [O] inform_proposed_resolution_15_days
```

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 Section 3. Note that we cannot compare to a baseline as either none of the related work provides the evaluation results or the task they addressed is a different one thus results are not comparable.

For measuring the effectiveness of the system, we evaluated the following outputs and we compared them to the outputs contained in the manual created gold standard: *i)* the number of correct sentences extracted from the text, *ii)* the number of correct terms identified in the extracted sentences, *iii)* the number of correct deontic components annotations performed on the identified terms, *iv)* the number of correct rules generated from the extracted terms annotated with the deontic component, and *v)* the impact of the CCG parser in supporting the detection of new patterns for generating rules.

By referring to the architecture described in Section 3, single blocks have been evaluated separately enabling a more fine-grained analysis of potential pitfalls in the effectiveness of system components. The evaluation of the lower branch of the pipeline consisted in the computation of the following values: *i)* the number of sentences extracted

correctly, *ii)* the number of terms detected, *iii)* the number of terms correctly annotated with the deontic tags, and *iv)* the number of rules that have been generated correctly. While, the evaluation of the upper branch, consisted in measuring: *i)* the agreement between the rules extracted from the CCG output and the ones generated by the lower branch, and *ii)* the number of rules correctly extracted from the CCG output regarding sentences for which the lower branch generated anything.

Lower Branch Evaluation The extraction of the sentences is 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 are 100%.

The second task is the identification of the terms within sentences. The gold standard contains 65 terms extracted by the analysts; our system is able to extract 59 terms whose 49 are correct. Therefore, the obtained recall is 90.78% and the precision is 83.05%, with a F-Measure of 86.74%. Concerning the assignment of the deontic annotation, 47 out of the 49 correctly identified terms have been annotated with the proper deontic component, leading to a precision of 95.92%.

The last step consists in determining which of the 36 rules contained in the gold standard have a counterpart in the automatically generated rule set composed by 41 rules. 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 with a Precision of 80.49% and a Recall of 91.67%.

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 rule set. This means that, as final assessment, we obtain a recall of Precision of 66.67%.

Upper Branch Evaluation The first evaluation performed on the upper branch of the pipeline was the measure of the agreement between the rules generated by the lower branch and the ones inferred from the output of the CCG parser. By starting from the output of the CCG parser, we firstly verify if words belonging to different terms are directly related by one of the logical modifiers used by the CCG parser for representing relationships between words. After the verification of the all generated relationships, we computed how many of them exist also in the set of the rules generated by the NLP parser used in the lower branch.

The set of relationships between terms extracted by the CCG parser contains 51 relationships and, by transforming them in rules, 35 of them have a counterpart (as defined in the previous paragraph) in the gold standard. With respect to the lower branch, the CCG parser was able to find 2 new rules having a counterpart in the gold standard. This means that the recall increased to 97.22% (35 rules detected out of 36), but the precision decreased to 68.63% due to the high number of relationships extracted by the CCG parser. Indeed, the CCG parser works at a more fine-grained logical-linguistic level with respect to the NLP-only parser; therefore, the detection of relationships between terms that are not actually a rule it is more easy.

Another interesting result is the number of deontic annotations that have been found by the CCG parser. The number of such annotations increased from 47 to 49 out of the 49 annotations contained in the gold standard by reaching a precision of 100.00%.

Finally, 29 out of the 35 rules found by the CCG parser have a full correspondence with the manually generated rules. This means that the synergistic use of the NLP-only and CCG parser led to an improvement of 5 rules by increasing the precision of the system from 66.67% to 80.56%.

Discussion of the Results As we mentioned above, the text we analyzed contains 35 prima facie clauses, and some of these rules require to be decomposed in two sub-rules to fully capture the nuances of the conditions under which the obligations hold. For example, as we have seen in Section 3, 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 norm could depend on the logical formalism used to reason with the rule. For example, if a condition of activation of an obligation is disjunctive, it is represented by two rules in the manually generated rule set. However, the disjunction could be represented by a single proposition encoding both of them. Thus, the number of rules required to model a normative clause depends on the underlying logic. This means that we can take as reference for the computation of the recall not the actual number of rules in the reference rule set, but the number of prima-facie clauses. In this case, the extracted rules cover 31 out of the 35 prima facie clauses. Finally, note that the examples used in this section (Section 8.2.1.a(vii) and 8.2.1a(xiii) are correctly identified by our system. Concerning error analysis, in most cases incorrect rules depend on the incorrect identification of the propositions in the first step, or on the fact that the rules contain implicit terms in the left-hand side to be derived from the right hand side. The correct treatment of these cases is left as future research.

5 Concluding remarks

In this paper, we have presented a combined framework for the automated extraction of rules from legal texts. The framework exploits both syntax-based patterns extracted using the Stanford Parser, and logic-based ones extracted through the Boxer framework. Several steps need to be addressed as future research to improve the performance and the applicability of the system. First of all, we need to construct a gold standard of legal rules extracted from different kinds of legal texts in order to validate the proposed approach on a larger dataset, taking into account the variability of the legal documents. Second, 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 [...]`. Then, 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. Third, 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.

References

1. van Engers, T., van Gog, R., Sayah, K.: A case study on automated norm extraction. In: Proc. of JURIX. (2004) 49–58
2. Wyner, A., Peters, W.: On rule extraction from regulations. In: JURIX-2011. (2011) 113–122
3. Curran, J.R., Clark, S., Bos, J.: Linguistically motivated large-scale NLP with c&c and boxer. In Carroll, J.A., van den Bosch, A., Zaenen, A., eds.: ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic, The Association for Computational Linguistics (2007)
4. Soria, C., Bartolini, R., Lenci, A., Montemagni, S., Pirrelli, V.: Automatic extraction of semantics in law documents. In: European Press Academic Publishing. (2005)
5. Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S., Soria, C.: Automatic semantics extraction in law documents. In: ICAIL-2015. (2005) 133–140
6. de Araujo, D.A., Rigo, S., Muller, C., de Oliveira Chishman, R.L.: Automatic information extraction from texts with inference and linguistic knowledge acquisition rules. In: Web Intelligence/IAT Workshops. (2013) 151–154
7. Kiyavitskaya, N., Zeni, N., Breaux, T.D., Antón, A.I., Cordy, J.R., Mich, L., Mylopoulos, J.: Automating the extraction of rights and obligations for regulatory compliance. In: ER-2008. (2008) 154–168
8. de Maat, E., Winkels, R.: Suggesting model fragments for sentences in dutch laws. In: Proc. of LOAIT. (2010) 19–28
9. Brighi, R., Palmirani, M.: Legal text analysis of the modification provisions: a pattern oriented approach. In: ICAIL-2009. (2009) 238–239
10. Francesconi, E.: Legal rules learning based on a semantic model for legislation. In: Proc. of SPLeT Workshop. (2010)
11. Boella, G., Caro, L.D., Robaldo, L.: Semantic relation extraction from legislative text using generalized syntactic dependencies and support vector machines. In: RuleML-2013. (2013) 218–225
12. Dragoni, M., Governatori, G., Villata, S.: Automated rules generation from natural language legal texts. In: ICAIL 2015 Workshop on Automated Detection, Extraction and Analysis of Semantic Information in Legal Texts. (2015)
13. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA (1998)