On Rule Extraction from Regulations

Adam WYNER [a,1] and Wim PETERS [b]

[a] *Department of Computer Science, University of Liverpool, UK.*

[b] *Department of Computer Science, University of Sheffield, UK.*

Abstract. Rules in regulations such as found in the US Federal Code of Regulations can be expressed using conditional and deontic rules. Identifying and extracting such rules from the language of the source material would be useful for automating rulebook management and translating into an executable logic. The paper presents a linguistically-oriented, rule-based approach, which is in contrast to a machine learning approach. It outlines use cases, discusses the source materials, reviews the methodology, then provides initial results and future steps.

Keywords. Text analysis, Regulation, Conditionals, Deontic Logic

1. Introduction

One of the long term goals artificial intelligence and law has been to identify, extract, and formalise conditional or normative rules from legal source materials. In so doing, one could facilitate search of the law, automatic reasoning for consistency and inference, or public services to support compliance, legal access, and transparency. For instance, there are XMLs for legal materials making them available, searchable, and linkable on the Internet such as CEN MetaLex[2] along with national standards, for example, in the United States[3]. Legislation as logic programs has a long history [15,2]. There is current work on expressing legal rules in XML such as RuleML [11]. Oracle Policy Automation[4] is a commercially successful solution in which the source material is manually scoped, translated into a formal, executable language, then served with a natural language interface over the Internet allowing users to receive determinations.

However, identifying, extracting, and formalising the rules remains a highly knowledge and labour intensive task, creating a significant bottleneck between the *semantic content* of the source material, expressed in natural language, and computer-based, automatic use of that content. To address the bottleneck, natural language processing (NLP) techniques have been applied. One approach uses machine learning [7] for Dutch and [3,4,10] for Italian; such techniques classify documents or sentences rather than annotate contentful elements and their relationships. Moreover, as argued in [16], the techniques generate untraceable rationales for the classifications, which arguably makes them less likely to be adopted for legal applications. Other approaches use linguistically oriented

[1]Corresponding Author. E-mail: adam@wyner.info

[2]http://www.metalex.eu/

[3]http://www.gpo.gov/fdsys/

[4]http://www.oracle.com/technetwork/apps-tech/policy-automation/overview/index.html

parsing, though they are not highly developed [13] or not available for English [8]; they do not target deontic expressions, diathesis alternations, or use thematic roles, which we claim are in principle essential for a sufficient semantic characterisation. Well developed parsers such as the Stanford Parser [9] have not been evaluated against legal sources. Moreover, basic tools seem to be bespoke, use complex models, are not in wide circulation, and are difficult to modify.

In contrast, [17,16] develop an approach to support the identification and extraction of legal case factors that is rule-based, bottom-up, linguistically expressive, and uses the open source tool General Architecture for Text Engineering (GATE) [6]. Most importantly, and unlike previous systems, high level, complex semantic concepts are annotated. In this paper, we extend that approach to the identification and extraction of rules from regulations, in particular, conditional and deontic rules, specifying the antecedents, consequences, agents, themes, actions, and exceptions; we include rules with lists. The main novel contribution of this paper is that we identify and extract high level components of rules from regulations in English, applying and extending widely available, current NLP tools. We provide an open source, modular framework along with an explicit methodology and open source materials. Thus, the approach and analysis can reproduced, tested, and built on.

The paper is structured as follows. In section 2, we present a particular use case for our application along with the sample materials. We report our initial analysis of the sample materials and a proposed model in section 3. Our approach is presented in section 4, giving the method, modules, and sample output. We compare our results against a Gold Standard in section 5. In section 6, we make a range of concluding observations and mention topics for future research.

2. Use Case and Materials

To ground the research, we consider issues related to compliance.[5] Particular industries, e.g. banking, health service providers, must comply with the rules found in legislation and regulations, so must track the rules as they develop. Broadly speaking, there are three stages. In the first stage, the relevant regulatory rules for particular industries are identified and extracted from source regulation to create compliance rulebooks; these rules are related to the industrial or company compliance policies. In the second stage, compliance risks are identified, that is, the difference between a company's current policies and novel rules. In the third stage, the rule is made operational by translating it into the particular business rule applicable to the company, tracking where practice complies with the regulation, and reporting any issues that arise in compliance. For our purposes, the first stage is relevant - the identification and extraction of rules from source materials; it is a necessary stage in feeding later processes, such as the formalisation and operationalisation of the rules. The generation and maintenance of compliance rulebooks can be outsourced to a company that is responsible for tracking and extracting those rules relevant to each industry and from across a range of sources of law. The next step in the process would be to translate the rules into an excutable logic such as Oracle's Office Rules or LegalRuleML. However, the method of identification and extraction of the rules from the sources is currently a knowledge and labour intensive task; automated support

[5] Thanks to John Cyriac of *ComplianceTrack* for discussion. See: http://www.compliancetrack.com/

for the task could not only be more efficient, but allow more flexible expressions of the rules, say in an XML format that could be variously transformed.

For source materials, we have selected a passage from the US Code of Federal Regulations, US Food and Drug Administration, Department of Health and Human Services regulation for blood banks on testing requirements for communicable disease agents in human blood, Title 21 part 610 section 40.[6] This is a four page document of 1,777 words. Despite its size, the document offers much to consider as a starting point.

In the next section we discuss a range of observations about this material, highlighting those components that we should identify and extract. Each of these have clear linguistic expression that is apparent to able users of the language, but which is not machine readable without additional, machine readable annotation.

3. Initial Analysis and Model

In this section, we discuss our initial analysis of the source material and a model for deontic and conditional rules.

3.1. Initial Analysis

We applied the Stanford Parser to the source document in order to identify the linguistic characterisations of the target elements. The Stanford Parser (version 1.6.8) is a robust, well-developed, well-maintained parser for English. It uses a Probabilistic Context-Free Grammar which was trained on the Penn Treebank; the Penn Treebank is a corpus of manually parsed newspaper articles; the parser performs well on newspaper articles. However, when we submitted our source document to the Stanford Parser (as a GATE plugin), it failed to parse. The source was then divided into subportions, creating a small corpus of documents, until all the subportions were successfully parsed. The parser outputs a range of syntactic parses, sequences of words that form a grammatical phrase, as well as dependency information, relationships between phrases such as *subject* of a sentence, *object* of a preposition, and so on. The parser generates a number of alternative parses that can be investigated for grammatical information. Our summary observations are based on such information.

First, we consider the failure of the parser to generate output. In our view, there are four causes, tied to the underlying source material used to train the parser, and in particular, to structural differences between our regulatory materials and newspaper articles. While the observations need to be further supported, we found that regulatory texts:

- Have *long, complex sentences* of several coordinated clauses or subordinate clauses. Such clauses may have several alternative parses.
- Use *lists*, which use list punctuation, including enumerations, colons, etc. Such punctuation confounds tokenisation and sentence splitting, which are essential, initial processes.
- Use *references* that contain a mix of punctuation and alpha-numeric characters that confound tokenisation and sentence splitting.

[6] See in general: http://www.gpoaccess.gov/cfr/index.html The citation to the regulation is 21CFR610.40. Search for regulations in https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/cfrsearch.cfm

These elements alone may contribute to long parse times or failures to parse. The observations indicate that *for parsers to work successfully on legal materials, specialised preprocessors and training corpora must be developed*.

For parsed documents, we find a range of additional issues, where regulatory texts:

- Contain embedded exception clauses.
- Contain active and passive sentences.
- Have ambiguities from alternative parses of noun phrase and prepositional phrase.

Despite these issues, the parses can be inspected for constituents of interest, though this is a manual, knowledge intensive task. Among the constituents, we find: clauses for exceptions, deontic concepts, main verbs, negation, subjects and direct objects, and the structures of conditional sentences. In other words, the parse provides basic grammatical information that we can use to construct our automated tool.

Given these observations, our objective is to develop processing modules to annotate a set of relevant components from the source text. As in [16], the modules provide a semi-automatic, interactive tool that highlights relevant passages, thereby assisting the extraction process as well as supporting the development of a Gold Standard corpora. In addition, as we discuss later, where error is identified, the rules can easily be refined to more correctly analyse the data.

3.2. Model

To focus the research, we propose a model for analysis. For deontic rules, we have :

- Agent and theme, which are semantic roles that must be associated with noun phrases in grammatical (subject or object) roles in the sentence. These are used to account for active-passive alternations and identify the individual's relationship to the deontic concept.
- Deontic modals and verbs.
- Main verbs.
- Exception clauses, which may appear in lists.

In addition, we are also interested to identify:

- Conditional sentences along with their antecedents and consequences. Antecedents may appear in lists.

There are other elements that could be identified, e.g. negation, temporal phrases, pragmatic expressions, generic sentences expressing deontic concepts [7], and Hohfeldian rights and powers [14], among others. Each of these require use or development of specific, additional modules. For this study, we focus on our model of elements; our approach is extensible, so the model is not intended to be definitive.

Models for rule identification have been proposed [7,3,4,10]. Our model is simpler, targeted at the source material at hand, and appropriate to our use case. Unlike [10], we match linguistic constituents of the source material to elements of the model: the main verb is selected, not a nominalisation; noun phrases serve as the parties to the rule, not a derivative expression; the Hohfeldian counterparties are not identified; the analysis is not set in the context of a theory of sorts of legal provisions, for which we do not have clear textual evidence; and we do not here differentiate between *legal domain* and *domain* *independent* terminology. Important novelties of our approach are the use of thematic roles and the identification of exception phrases.

4. Method, Modules, and Sample Output

We have used GATE, which is an open-source framework for language engineering applications [6,16]. The interface enables linguists and text engineers to develop and apply a variety of natural language processing tools to a corpus. In section 4.1, we discuss our methodology, and in section 4.2, we outline the processing modules.

4.1. Method

As a methodological starting point, we adopt existing approaches to event recognition [5]. Event recognition is one of the major tasks within information extraction, which has been successfully applied in research areas such as bioinformatics, news aggregation, business intelligence and text classification. Recognising events is generally carried out using pre-defined templates or relations, where the slots are known and the values are entities extracted from the text. The task is then domain dependent, but feasible.

Our methodology has the following aspects [17,16]:

- Rule-based and unweighted: the analysis makes use of lookup lists (gazetteers) that ascribe base annotations to strings along with unweighted rules (using the Java Annotation Pattern Engine (JAPE)). The output is not probabalistic.
- Bottom-up from simpler to complex: rules make use of simpler annotations to construct more complex annotations.
- Linguistically well-founded: rules make use of syntactic analyses and lexical-semantic information.
- Semi-automatic, interactive, and collaborative: where several alternative analyses are possible, these are provided to a human user to highlight and select; the methodology supports collaborative annotation (e.g. GATE TeamWare).
- Modular, iterative, incremental development: the analysis takes into account a continuum of textual complexity and richness; modules are designed to address simpler data, tested, then revised to account for more complex data; where data is not accounted for, the reasons are pinpointed and used to revise the rules or otherwise left for future work; the lists and rules can be augmented and adapted.
- Transparent and reproducible: all materials can be examined and reused.
- Traceablity: results can be fully analysed in terms of lists and rules.

Alongside the development of processing modules, the methodology uses a range of forms of materials. We have the original source materials (Source). Yet, given the processing issues, it was divided into several subsections, each of which can be parsed (Source Sections) as discussed in section 3.1. Having identified issues with *long, complex sentences* as well as with *enumerations and references*, derivative documents were created that simplified over those confounding aspects (Source Derived). To develop particular modules, we created documents (Test Documents) containing only particular elements of the model (e.g. exceptions, deontic concepts, thematic roles, etc). Modules that work for the Test Documents are then applied to the Source Derived document, making observations or revisions. At this stage of the research and the observations in section 3.1, the resultant modules were not applied to the Source Sections or Source; however, this does not imply that our modules cannot apply to the Source material, but that we can *only test them pending resolution of the specific problems that are independent of our modules and must be addressed with specific tools*. Our examples below are shown with respect to Source Derived. Finally, we manually parsed the Source Derived document, producing a Gold Standard against which to compare the performance of our modules. In future work, development and evaluation texts will be distinct.

4.2. Processing Modules and Sample Output

The GATE platform [6] enables template-based extraction on the basis of heuristic pattern-based grammars as well as a pipeline of standard natural language processing components such as tokenisation, sentence splitting, part of speech tagging, morphological analysis (lemmatisation), verb and noun phrase chunking, and a parser. In this study, we used the Stanford Parser.[7] Using information from these components, we have created targeted annotation modules for elements of our model; these modules are expressed in the gazetteer lists and Java Annotation Patterns Engine (JAPE) rules.[8] In the following, we discuss the modules followed by examples.

To identify and extract elements, we define gazetteers and JAPE rules. A gazetteer is a list of strings that are associated with a central concept, which is represented as an annotation on the string; JAPE rules are used to create complex transductions, using annotations and regular expressions as input to produce annotations as output. Once a sequence of processes has been applied to a corpus, one can view annotations in documents or query for annotations [1]; alternatively, we can output results in XML.

We discuss together exceptions, list structures, and conditionals since exceptions and conditionals can both make use of lists; we provide a screenshot of a sample output. We have gazetteers and JAPE rules for various terms for *exception*, e.g. *except*, *unless*, *other than*, and others; a term on this list is annotated *Exception*. We create a JAPE rule for exception clauses, consisting of an Exception annotation followed by one out of a set of alternative syntactic phrases such as prepositional phrase (PP), noun phrase (NP), nominal phrase (NN), and sentence (S), which are indicated by the Stanford Parser. The sequence is annotated as *ExceptionPhrase2*. This rule shows how we can annotate the text with higher level semantic annotations from lower level annotations.

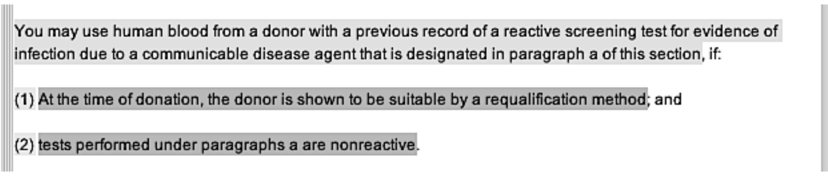
```
( {Exception}
({SyntaxTreeNode.cat==PP} | {SyntaxTreeNode.cat==NP | {SyntaxTreeNode.cat==NN} |
{SyntaxTreeNode.cat==S})) :temp
–> :temp.ExceptionClause2 = {rule = "ExceptionClause2"}
```

The identification of list structures is complex, requiring the identification of list markers of alternative forms such as punctuation and list labels, indicators that what follows is a list, along with first, midde, and last elements of the list. Given such a module, we can then identify lists per se as well as lists which follow indicators for antecedents and for exceptions.

---

[7] While it would have been attractive to use C&C/Boxer in GATE, it is not yet available as a plugin; other parsers are available in GATE (OpenNLP and LingPipe), but it is not the purpose of this study to cross-compare parsers.

[8] The materials and application are available upon request from the authors.

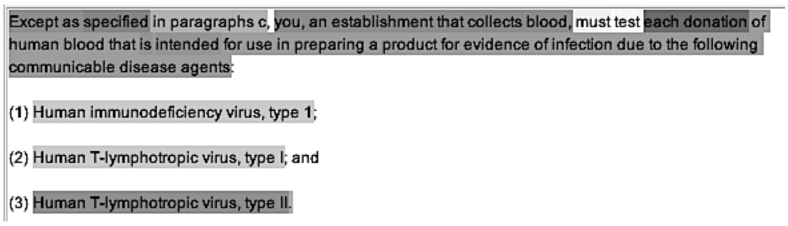


You may use human blood from a donor with a previous record of a reactive screening test for evidence of infection due to a communicable disease agent that is designated in paragraph a of this section, if:

(1) At the time of donation, the donor is shown to be suitable by a requalification method; and

(2) tests performed under paragraphs a are nonreactive.

Figure 1. Conditional with List Antecedent



Except as specified in paragraphs c, you, an establishment that collects blood, must test each donation of human blood that is intended for use in preparing a product for evidence of infection due to the following communicable disease agents:

(1) Human immunodeficiency virus, type 1;

(2) Human T-lymphotropic virus, type I; and

(3) Human T-lymphotropic virus, type II.

Figure 2. Deontic Rule with Exception Clause and List

Conditional sentences are those which contain the conditional marker *if*; they appear in alternative forms e.g. *If Bill is happy, Jill is happy*; *If Bill is happy, then Jill is happy*; *Jill is happy, if Bill is happy*; *Jill is happy if: (a) Bill is happy; and (b) Bill and Jill are together.* Additional rules are required to handle other contexts (e.g. with *then* and lists). They contain antecedents (that can be lists) and consequents. See Figure 1 (GATE produces coloured output, so the black and white figure is only indicative).

For deontic rules, we identify the deontic operator, the main verb, the semantic roles that noun phrases play, and any exceptions. For the deontic concepts, we have a gazetteer for each basic deontic concept, e.g. *Obligation*, *Prohibition*, and *Permission*, where each is a list of terms that synonymously express that concept; for *Obligation*, we have *must*, *obligate*, *obligation*; tokens in the text are lemmatised and matched against the gazetteers. The Stanford Parser annotates the main verb *Verb*.

A linguistically interesting module is the mapping of thematic roles to syntactic position given alternative syntactic patterns (*diathesis alternations*) such as the *active-passive alternation*: in the active sentence, *You must label each donation*, the agent of the action *you* is found in the subject position and the theme *each donation* is in the object position; in the corresponding passive sentence, *Each donation must be labelled by you*, the theme is in the subject position, while the agent is in a *by-phrase* or is implicit. For deontic notions, it is essential to identify the agent of the action, not simply the subject of the sentence, as the bearer of the obligation. To associate grammatical roles (subject and object) with thematic roles (agent and theme), we use grammatical information from the Stanford Parser (passive annotation and dependency information) along with information on thematic roles derived from VerbNet [12,16].

In Figure 2, we present a deontic rule, where the annotations in order are: an exception clause *Except as specified in paragraph c*; an agent *you, an establishment that collects blood* in subject position (the sentence is active), the modal operator *must*, the main verb *test*, the theme *each donation of human blood....agents* in object position, and a list of elements (1)-(3). In Figure 3, we have a passive deontic rule.

Required testing must be performed by a laboratory registered in accordance with part 607 of this chapter and either certified to perform such testing on human specimens under the Clinical Laboratory Improvement Amendments of 1988.

Figure 3. Deontic Rule with Passive

5. Results

Comparing the results of our automated modules against our Gold Standard, we have the precision and recall results in Table 1. Several of the results are ideal, while others are poor. The ideal results for the identification of the deontic concepts ought not to be surprising since it requires only lexical lookup from an unambiguous list of items with with few morphosyntactic variants (e.g. present and past tense), and furthermore, there are no dependencies. List identification is ideal for this data, but after all, the module was written to fit the data; nonetheless, it shows we have an analysis, which can be further developed. Similar comments apply to the identification of antecedents and consequents. That said, the rules for such identifications are not straightforward since there are dependencies between elements, so the results are not trivial.

The other results present more interesting linguistic issues. The precision for exception phrases is not surprising in the sense that identification of the phrases depend on identification of the head of the exception phrase, which itself is lexical lookup from an unambiguous list. In contrast, the poor recall indicates that the dependent phrasal complements for these heads have yet to be correctly identified in the rules.

Mapping thematic roles (agent and theme) to syntactic positions (subject, object, and by-phrase) raise several issues. First, we consider active sentences. For agentive subjects, precision results show that the Gold Standard identifies subject nouns that are not agents, while the rules identify them as agents. This rests on the polysemy of several verbs - *designate*, *apply*, and *contain* - that have agentive and non-agentive senses; at this point, the rules do not distinguish senses. For theme direct objects, the precision is high, but the recall is low: where a noun phrase that immediately follows a verb with a theme role, the noun phrase is likely to have the theme role; in contrast, we find that dependencies are not accurately parsed where, for example, prepositional phrases are incorrectly left out of the noun phrase, subordinate clauses are incorrectly included in the noun phrase, phrases intervene between the verb and direct object, or direct objects are not marked with a theme role. The first two issues bear on the parser, the second could be addressed with augmented rules, while the latter may be a matter of using other thematic roles.

The passive sentences have similar issues for active sentences so we focus remarks on different problems. Results are filtered first with respect to the passive auxiliary, so the precision results make sense. The recall results are lower. Recall for passive is itself low; examination of the data shows that this is largely due to relative clauses with elided passives as in *a lab registered....* rather than *a lab that has been registered*. Another reason is where we find conjoined passive clauses. Missing such passives is a reason behind the poor recall for agents and themes in passivised sentences. Notice, finally, that recall for subjects is better than for non-subjects; this is founded in a linguistic observation that there are fewer additional linguistic elements that typically intervene between a subject and verb than between a non-subject and verb. This highlights that the rules are sensitive to relative position rather than semantic or syntactic role.

Table 1. Results over Source Document

Category	Precision	Recall
Obligation	100%	100%
Permission	100%	100%
List	100%	100%
Antecedent	100%	100%
Consequent	100%	100%
Exception Phrases	100%	36%
Subject Agent	88%	100%
Direct Object Theme	100%	30%
Passivised Verb	100%	57%
Subject Theme	100%	70%
By-Phrase Agent	100%	14%

6. Next Steps

Some of these preliminary results are highly promising, while other results indicate a need for further work. We have emphasised that the tool is intended to be used not only to identify and extract relevant elements, but also to help pinpoint those aspects of the analysis which need further development. As we have mentioned out, there are independent confounding issues bearing on coordination, subordination, and prepositional phrase attachment; for example, the Stanford Parser does not reliably parse prepositional phrase or subordinate clauses with respect to noun phrases. Our analysis highlights a range of additional parsing issues concerning relative clauses, negations, infinitivals, adverbial phrases, and so on. There are issues concerning semantic scope of negations and *exception* clauses as well as mappings of thematic roles to syntactic positions.

Despite some of the poor results and additional complexities, there is an important advantage in our rule-based, linguistically oriented approach, for each of the issues identified above can be tied to some particular lexical, syntactic, or semantic issue. We expect that modularising the task to target the problems, then combining the results, will lead to better overall results. The analysis also supports in depth investigation into the creation of the Gold Standard text. The process of analysis would be further supported with the creation of a large scale corpus of parsed legal documents.

To some extent, the issues we have analysed are generic to the textual analysis. What makes this work particularly relevant to AI and Law is the identification of rules, especially deontic rules, as well as of some confounding features of regulatory text that must be addressed in order to continue to make progress.

Finally, we have the building blocks to identify components of rules that are useful for extracting information to a rulebook as well as to map to a executable logical language. In addition to deepening our investigation, we plan develop mappings to these external sources for those fragments that have been successfully identified.

Acknowledgements

The first author was supported by the FP7-ICT-2009-4 Programme, IMPACT Project, Grant Agreement Number 247228. The views expressed are those of the authors.

References

[1] Niraj Aswani, Valentin Tablan, Kalina Bontcheva, and Hamish Cunningham. Indexing and querying linguistic metadata and document content. In *Proceedings of 5th International Conference on Recent Advances in Natural Language Processing*, Borovets, Bulgaria, 2005.

[2] Trevor Bench-Capon, George Robinson, Tom Routen, and Marek Sergot. Logic programming for large scale applications in law: A formalisation of supplementary benefit legislation. In *International Conference on Artificial Intelligence and Law*, pages 190–198, 1987.

[3] C. Biagioli, E. Francesconi, A. Passerini, S. Montemagni, and C. Soria. Automatic semantics extraction in law documents. In *Proceedings of the 10th international conference on Artificial intelligence and law*, ICAIL '05, pages 133–140, New York, NY, USA, 2005. ACM.

[4] Raffaella Brighi and Monica Palmirani. Legal text analysis of the modification provisions: a pattern oriented approach. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, pages 238–239, New York, NY, USA, 2009. ACM.

[5] Nancy Chinchor, Lynette Hirschman, and David D. Lewis. Evaluating message understanding systems: An analysis of the third message understanding conference (muc-3). *Computational Linguistics*, 19(3):409–449, 1993.

[6] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, pages 168–175, 2002.

[7] Emile de Maat and Radboud Winkels. Automated classification of norms in sources of law. In Enrico Francesconi, Simonetta Montemagni, Wim Peters, and Daniela Tiscornia, editors, *Semantic Processing of Legal Texts*, volume 6036 of *Lecture Notes in Computer Science*, pages 170–191. Springer Berlin / Heidelberg, 2010.

[8] Emile de Maat and Radboud Winkels. Suggesting model fragments for sentences in dutch laws. In *Proceedings of Legal Ontologies and Artificial Intelligence Techniques (LOAIT 2010)*, pages 19–28, 2010.

[9] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of Language Resources and Evaluation Conference (LREC 2006)*, 2006.

[10] Enrico Francesconi. Legal rules learning based on a semantic model for legislation. In *Proceedings of the LREC 2010 Workshop on the Semantic Processing of Legal Texts (SPLeT-2010)*, Malta, May 2010.

[11] Guido Governatori. Representing business contracts in RuleML. *International Journal of Cooperative Information Systems*, 14(2-3):181–216, 2005.

[12] Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. A large-scale classification of english verbs. *Language Resources and Evaluation*, 42(1):21–40, 2008.

[13] L. Thorne McCarty. Deep semantic interpretations of legal texts. In *ICAIL '07: Proceedings of the 11th International Conference on Artificial Intelligence and Law*, pages 217–224, New York, NY, USA, 2007. ACM Press.

[14] Hamish Ross. Hohfeld and the analysis of rights. In James Penner, David Schiff, and Richard Nobles, editors, *Introduction to Legal Theory and Jurisprudence: Commentary and Materials*, pages 595–647. Buttersworth Law, London, 2002.

[15] Marek Sergot, Fariba Sadri, Robert Kowalski, Frank Kriwaczek, Peter Hammond, and Therese Cory. The british nationality act as a logic program. *Communications of the ACM*, 29(5):370–386, 1986.

[16] Adam Wyner and Wim Peters. Lexical semantics and expert legal knowledge towards the identification of legal case factors. In Radboud Winkels, editor, *Proceedings of Legal Knowledge and Information Systems (JURIX 2010)*, pages 127–136. IOS Press, 2010.

[17] Adam Wyner and Wim Peters. Towards annotating and extracting textual legal case factors. In *Proceedings of the Language Resources and Evaluation Conference Workshop on Semantic Processing of Legal Texts*, Malta, 2010. To appear.