Workshop Programme

14:30-14:45 Welcome and introduction

14:45-15:10 *A Description Language for Content Zones of German Court Decisions*
Florian Kuhn

15:10-15:35 *Controlling the language of statutes and regulations for semantic processing*
Stefan Hoefler and Alexandra Bünzli

15:35-16:00 *Named entity recognition in the legal domain for ontology population*
Mírian Bruckschen, Caio Northfleet, Douglas da Silva, Paulo Bridi, Roger Granada, Renata Vieira, Prasad Rao and Tomas Sander

16:00-16:30 Coffee break

16:30-16:55 *Legal Claim Identification: Information Extraction with Hierarchically Labeled Data*
Mihai Surdeanu, Ramesh Nallapati and Christopher Manning

16:55-17:20 *On the Extraction of Decisions and Contributions from Summaries of French Legal IT Contract Cases*
Manuel Maarek

17:20-17:45 *Towards Annotating and Extracting Textual Legal Case Factors*
Adam Wyner and Wim Peters

17:45-18:10 *Legal Rules Learning based on a Semantic Model for Legislation*
Enrico Francesconi

Workshop Organisers

Enrico Francesconi (Institute of Legal Information Theory and Techniques, CNR, Italy)
Simonetta Montemagni (Istituto di Linguistica Computazionale, CNR, Italy)
Wim Peters (Natural Language Processing Research Group, University of Sheffield, UK)
Adam Wyner (Department of Computer Science, University College London, UK)

Programme Committee

Table of Contents

Preface v

A Description Language for Content Zones of German Court Decisions 1
Florian Kuhn

Controlling the language of statutes and regulations for semantic processing 8
Stefan Hoefler and Alexandra Bünzli

Named entity recognition in the legal domain for ontology population 16
Mírian Bruckschen, Caio Northfleet, Douglas da Silva, Paulo Bridi, Roger Granada, Renata Vieira, Prasad Rao and Tomas Sander

Legal Claim Identification: Information Extraction with Hierarchically Labeled Data 22
Mihai Surdeanu, Ramesh Nallapati and Christopher Manning

On the Extraction of Decisions and Contributions from Summaries of French Legal IT Contract Cases 30
Manuel Maarek

Towards Annotating and Extracting Textual Legal Case Factors 36
Adam Wyner and Wim Peters

Legal Rules Learning based on a Semantic Model for Legislation 46
Enrico Francesconi

Author Index

Paulo Bridi	16
Mírian Bruckschen	16
Alexandra Bünzli	8
Douglas da Silva	16
Enrico Francesconi	46
Roger Granada	16
Stefan Hoefler	8
Florian Kuhn	1
Manuel Maarek	30
Christopher Manning	22
Ramesh Nallapati	22
Caio Northfleet	16
Wim Peters	36
Prasad Rao	16
Tomas Sander	16
Mihai Surdeanu	22
Renata Vieira	16
Adam Wyner	36

Preface

The last few years have seen a growing body of research and practice in the field of Artificial Intelligence and Law on the automated processing of legal information including: legal reasoning and argumentation, semantic and cross-linguistic legal information retrieval, document classification, legal drafting, legal knowledge discovery and extraction, as well as the construction of legal ontologies and their application to the legal domain. It is of paramount importance to use Natural Language Processing techniques and tools to automate knowledge extraction from legal texts, which are expressed in natural language.

Over the last two years, there have been a number of dedicated workshops and tutorials on different aspects of semantic processing of legal texts: the LREC 2008 Workshop "Semantic Processing of Legal Texts", the JURIX 2008 Workshop "The Natural Language Engineering of Legal Argumentation: Language, Logic, and Computation (NaLEA)", the ICAIL 2009 Workshop "The 3rd Workshop on Legal Ontologies and Artificial Intelligence Techniques joint with the 2nd Workshop on Semantic Processing of Legal Texts", and the ICAIL 2009 Workshop "The Natural Language Engineering of Legal Argumentation: Language, Logic, and Computation".

To continue this momentum, a 3rd Workshop on "Semantic Processing of Legal Texts" was organised at the LREC 2010 conference to bring to the attention of the broader language resources and human languages technology community the motivations, objectives, and technical challenges posed by the semantic processing of legal texts. The outcome of these interactions are expected to advance research and applications and foster interdisciplinary collaboration within the legal domain.

The main goals of the workshop are to provide an overview of the state-of-the-art in legal knowledge extraction and management, to explore new research and development directions and emerging trends, and to exchange information regarding language resources and human languages technologies and their applications to the legal domain.

Seven papers were accepted for presentation to the workshop. In brief, the topics of the papers are as follows. Kuhn outlines the linguistic features of German court decisions so as to support the automatic analysis of the decisions. Hoefler and Bunzli describe the development and application of a controlled language of German for the semantic processing of Swiss statutes and regulations. Bruckschen et al. motivate and present an approach to the population of a legal ontology using natural language processing to identify the relevant entities; they discuss some experimental results. Surdeanu et al. introduce an approach to extract fine-grained information (e.g. patents and laws) from designated segments of text (e.g. claims) rather than the whole text. Maarek discusses automated extraction of decisions from summaries of French legal IT contract cases. Wyner and Peters present a methodology for the automated annotation of legal case factors in a common law setting. Francesconi applies NLP techniques and machine learning to extract legal rules from legislative texts.

We would like to thank all the authors for submitting their research and the members of the Program Committee for their careful reviews and useful suggestions to the authors. We also would like to thank the LREC 2010 Organising Committee that made this workshop possible.

Workshop Chairs

Enrico Francesconi
Simonetta Montemagni
Wim Peters
Adam Wyner

A Description Language for Content Zones of German Court Decisions

Florian Kuhn

Cognitive Science Center of Excellence / Dept. of Linguistics
University of Potsdam
Karl-Liebknecht-Str. 24-25, 14476 Potsdam, Germany
fkuhn@uni-potsdam.de

Abstract

d We present a work-in-progress report of our research on automatically analyzing German court decisions. A description language for linguistic features in content zones of a court decision is introduced, developed to cover linguistic features of German court decisions. We motivated our research with significant text characteristics found in our corpus of private law decisions and show how we map these characteristics to elements of the description language. Finally, further research aspects are mentioned.

1. Introduction

While many text domains have a quite similar usage of language, the judicial language uses semantics and interpretation in a unique manner which makes it less transparent for laymen to comprehend.
In the German legal domain, these characteristics created the discipline of German legal linguistics, discussing characteristics of text production and interpretation. Although there is a long tradition of legal text production and interpretation, research by linguists is relatively young and spread with growing popularity of text linguistics and its models of interpretation and argumentation (Busse, 2000).
Influence of computational techniques and computer science to the legal domain gave birth to *legal informatics*, a discipline in between computer science and law, dealing with legal information retrieval and application design for law experts but also with expertise on IT-related issues like copyright and security.

However, contact between German legal linguists and computer science has been sparse and still offers many opportunities.
A discipline suitable for linking both legal linguistics and computer science is *computational linguistics*. This interdisciplinary field is dealing with description of linguistic theories by means of formal methods derived from computer science, including rule-based and statistical approaches. Research topics of computational linguistics are, for example, machine translation, natural language parsing, statistical corpus linguistics, speech synthesis and speech recognition.
The work in progress presented in this paper ties up to computational linguistic research that has been done in the legal domain, ranging from structure analysis (Moens et al., 1996) to argument analysis (Mochales Palau and Moens, 2009) and the extraction of judicial definitions (Walter, 2008)
This paper will focus on content description and structuring of German private law court decisions. The observations and description language developed and shown is intended to be used as important requirement for our future research that will concentrate on argument analysis of German private law decisions.
Related work will be found in the next section, where a first interdisciplinary approach as well as aspects of legal information retrieval and computational linguistic methods are shown to emphasize the different aims and methods of research.
In the third section, our corpus of German private law decisions is introduced. Several considerations like documenth length, source variability and subjects are mentioned.
Based on observations made on the corpus and prescrioption made in German law, we then define a document structure divided as by so called content zones. Hierarchical and logical restrictions of this 'text grammar' are also considered.
Section four introduces linguistic features found in the documents of the corpus and discusses how to integrate them into the provided zone structure, telling between zones that are defined by one feature and such that require more complex linguistic aspects to be describend probably.
The observations made are then used to develope a description language that defines the genre of German private law decisions by means of (text) linguistic categories. This is done in section six.
Because the paper refers to work still in progress, consideration and discussion of future research plans are

to be found in the last section of this paper.

2. Related Work

2.1. Legal Informatics

Legal informatics is a branch of applied computer science covering law related tasks. Legal information retrieval is one of the main research topics of legal informatics. A survey on legal information retrieval can be found in (Schweighofer, 1999), where the evolution of machine learning approaches in this discipline are described. Certain information retrieval methods like maximum entropy models and support vector machines are often used for aspects of computational linguistic approaches like (Mochales Palau and Moens, 2009).
Another important research topic of legal informatics is document management including format standardization for web-based applications. A survey on this large research field can be found in (Biasiotti et al., 2008).

2.2. Automated Legal Document Summarization

First interdisciplinary attempts of automated legal document processing for German language go back to the 1970-1974 work group *Interdisziplinäre Arbeitsgruppe 'Analyse der juristischen Sprache'*[1] founded by the DFG[2](Busse, 2000). The project's aim was to automatically paraphrase German legal texts. Law experts, computer scientists and text linguists took part. Even though the project did not succeed according to (Busse, 2000), it is remarkebable for being an early and large-scaled interdisciplinary enterprise.

An early approach to statistical and text grammar driven legal content analysis is the SALOMON project (Moens et al., 1996) for structure analysis of Dutch criminal cases. The authors used a text grammar to identify and summarize Dutch criminal cases for the purpose of guidance to lawyers. In two steps, relevant text units were extracted. First, a SGML-syntax text grammar identified the case structure. Then, shallow statistical methods retrieved text segments of the alleged offenses and the opinion of the court. The system performed well with best precision values about 82% and 95% depending on the methods.

Another project to summarize legal documents is the SUM project (Hachey and Grover, 2006) summarizing an XML-corpus of judgments of the *House of Lords*. A rhetorical annotation scheme inspired by (Teufel and Moens, 2002) was utilized here. Moreover, automated linguistic annotation was done by processing the documents with complex tokenizer (including part of speech tagging) and a linguistic analysis (including named entity recognition and clause detection). After annotation, a number of well-established machine learning techniques were used to train rhetorical role and relevance classifiers. Experiments based on cue phrase information, Support-Vector Machines and Maximum Entropy showed encouraging results, which motivated the authors to conclude that these steps may underlie the subsequent summarization algorithm as well as highlighting the utility of the rhetorical annotation scheme for legal discourse, thus its relevance.

2.3. Definition Extraction

One of the works on German law language we are aware of is (Walter and Pinkal, 2005). Here, the idea is to extract legal *definitions* by using a rule-based approach, working with a corpus of some 6000 verdicts of German environmental law. The verdicts were first processed by a dependency parser to construct abstract semantic representations of the sentences. These were used to transform the definition's dependency patterns into a set of 33 extraction rules. Different evaluation techniques were used, and the best precision values achieved were slightly above 70%.

2.4. Argument Mining

Recent research on detection, classification and structure of arguments in the legal domain has been done in (Mochales Palau and Moens, 2009). For this task of *argument mining*, the authors first consider several preliminaries based on representations of argumentation schemes by (Walton et al., 2008) and a number of discourse theory works. They then developed a formalism valid for argument mining covering elementary units of argumentation, their internal structure and the relations between them. For the detection of arguments, naive Bayes, maximum entropy model and support vector machine techniques were used. In a second phase, the detected arguments are classified according to their proposition (premises and conclusion), first parsing clauses and secondly statistically classifying the clauses with sophisticated features. In the last sub-task, the authors analyzed the argumentation structure by means of a context-free grammar and manually derived rules. For evaluation purpose, two corpora were obtained (*Araucaria Corpus* and Texts from the *European Court of Human*

[1]Interdisciplinary work group 'Analysis of legal language'.

[2]Deutsche Forschungsgemeinschaft (German Society for Research).

Rights). Best results for clause classification reached around 68 % and 74 %. Using the CFG-method for detecting argumentation structure, about 60 % accurracy and 70 % f-measure were scored. This work shows that argumentation mining using elements both of argument an discourse theory with statistical classification methods in a linguistically more sophisticated framework are promising for further research.

Finally, aspects of the content zone concept for German court decisions as well as first frequency analysis results were shown by us in (Stede and Kuhn, 2009a) and (Stede and Kuhn, 2009b).

3. Defining Content Zones

We restricted our view on German private law decisions. While looking at a number of texts issued by different county courts we discovered frequent similarities in terms of overall structure of these documents.
We then collected a small corpus of 40 German private law decisions of 12 different county courts, trying to cover many legal domains and documents of different length between approximately four and 13 pages. All All documents were annotated using a document structure schema described in the next section.

The general sequence of this structure is prescribed in the German *Zivilprozessordnung (ZPO)* but differs in detail. These differences for include the order of information or certain diction employed at particular positions in the text. With this regularity of segments at hand, we postulated their role as *content zones* similar to (Bieler et al., 2007), which idea of content zones is based on *move analysis* (Swales, 1990).
Content zones are text areas that carry significant information for the genre. They are arranged to occur in identical order in the text in most cases, however sometimes as optional element in a document. Often, identification of these zones is possible via keyword search or similar formal features. In some cases, however, it is helpful to analyze linguistic structures or the position of a zone in relation to other.
We also made use of the two type concept of zones in (Bieler et al., 2007) and applied formal and functional ones. In legal context, formal zones contain information on the document and the circumstances of the trial. This can be file identifier, names, dates and fixed phrases and parts of the summary of the judgement. Functional zones are mainly represented by point of view, argumentation and opinion.

We furthermore introduced a hierarchical discrimination of zones: More general zones are called *global zones*, while more fine-grained zones, which are included in the general ones and contain the . In a commmon German court decision, they are represented by the four main sections of *caption*, *summary*, *facts* and *justification*. A complete sequence of global zones thus defines a document of a genre. Global zones can be optional and of functional or formal type. They contain a more special second hierarchical type of content zone: *local zones*. A sequence of certain local zones defines a global zone. They are exhaustive so there is no segment not covered by them. Like global zones, local zones can be arranged in sequence or optional elements as well. They contain text segments important for the coherence structure of the document. Both types, global and local zones, vary in text length in relation to the document's overall length, but it is obvious that *facts* and *justification* deliver the main portion of a document since they develop information vital to jurisprudence. Table 1 shows a summary of global zones and the local zones they contain.
Note that headers found in *caption*, *facts* and *justification* are also treated as local zones.

Tag	Definition/ Local Zones
`caption`	header, court-name, case-identifier, date, plaintiff, defendant, formulae.
`summary`	Consequences both for plaintiff and defendant
`facts`	header, general description, plaintiff's view, plaintiff's proposition, defendant's view, defendant's proposition
`justification`	header, Introductory statement, Subsumption, Secondary judgement

Table 1: Global and local zones.

Based on the inventory of zones, we developed an XML-Schema representation to describe the surface structure of a German court decision. It is used to validate the document structure prior to any further analysis.

4. Content Zones and Linguistic Features

In a comparative study of German and Danish private law court decisions (Engberg, 1992), analyzed the relations between text conventions of the genre and

the speech acts[3] performed. He found some conventions to be *coded* to signal certain speech acts. For example in the German complaint and demurrer of the decision, he noted key verb phrases like *behaupten* (to claim) and *der Meinung sein* (to hold that). or *geltend machen* (to argue) introduced with either mentioning of *Kläger* (plaintiff) or *Beklagter* (defendant), and thus stating the argument for one party. However, the verb phrases alone do not have enough significance to categorize a speech act because there is a great variety of such phrases. This is where sentence mode, in this case subjunctive, helps to signal the according speech act. Interestingly, this is not true for the Danish decisions, where a set of key verb phrases is sufficient to signal arguments in complaint or demurrer while subjunctive is not used at all.
For our work, those observations seem very relevant since they show that there is certain regularity of linguistic categories that correlate with text function aspects.

To identify global and local zones and both their modes (formal and functional) we have to consider linguistic information. Henceforth, this information will be called *features*.
By having a closer look at court decision documents, it quickly becomes clear that they comprise many keyword phrases. These features rarely alter. In some cases, like *headings*, they even have a fixed position in relation to other text elements. Often there are also wordings that mark a formal declaration like *Im Namen des Volkes* (*In the name of the people*). Such features are not only fixed in the document itself but also principally unchangeable in all court decisions of that type. However, we also find keywords like names, dates and filenames that do change. The fact that zones map to at least one linguistic features to maintain significance can be named a *zone-feature relation zfr*.

4.1. Minimal Zone-Feature Relations

In minimal cases, a zone is defined by exactly the very keyword phrase that contains all of its content, and actually such zones often occur according to our definition. For example, either all header-zones that start a global zone or all wordings found are minimal *zfr*s. All of these minimal zone-feature relations are implied by local zones and in general are formal type. Most of them occur at the beginning of a court decision when concise information on trial and parties is relevant. These are *court name*, *names of judges*, *plaintiff* and *defendant* etc.
Others are distributed over the document, like headers or wordings. They both mark endings or beginnings of new sections in the text. This function also restricts the place of their appearance as said above. For example, a header like *Tatbestand* (facts) definitely ends the judgement summary zone and starts the facts zone. For wordings, we find phrases such as *hat ... für Recht erkannt* (... acknowledged (the right)) which in our corpus of 40 documents frequently start the summary zone.
Because zones with minimal *zfr* just bear one feature, the feature included is always *sufficient* for positive zone identification. However, this does not mean its occurrence is always mandatory, so there can also be zones that have a sufficient feature but are optional nonetheless.

4.2. Complex Zone-Feature Relations

Although keyword phrase matching is a common strategy to find certain content, there are also many segments in a texts that need more linguistic knowledge to be identified. Even though in many cases some keyword is rather sufficient to identify its corresponding zone once it is found, there might be a grammatical feature, for example, which supports to classify the content zone. Zones that use more complex feature relations are found in the functional global zones of *facts* and *justification* where the different views on the subject are developed by the different parties. The most simple type of complex zone feature relations is a set of keywords that defines this zone, for example *nebenentscheidungen*(secondary judgement) or *anlagenbezug* (attachment reference). However, there are complex types that contain more linguistic information than just several keyword phrases. They also cover syntactic aspects of the text that are helpful to identify the zone. Syntactic features we took into account are, for example, *tense*, *active/ passive*, *indicative/ subjunctive mood* and *adverb constructions*. Another very important feature are *connectives* like *conjunctions*, *subjunctions* and *adverbs*. Because they are frequently used as cohesive elements, they also occur in court decisions to support argumentation.
Complex zone-feature relations do not occur in formal global zones like *rubrum* (introduction), because a formal zone lacks syntactic complexity. Text in the zone *tenor* (summary) often utilizes extended infinitive and passive constructions to signal the consequences for the defendent/plaintiff, however complexity is very limited compared to the argumentations developed in *tatbestand*(facts) and *entscheidungsbegründung* (justification).

[3]The following example is restricted to argumentation of the parties involved.

In the *facts* zone, we find significant changes on syntactical level that help to differ between the zones that describe the view of every party: While undisputed facts are grammatically *declarative*, the sentence mode changes to *subjunctive* when depicting one of the disputable views. This feature is supported by the frequent use of predicative nouns like *der Meinung/Ansicht sein* (to have the opinion that) or *meinen* (to think that) and a reference to plaintiff/ defendant.
When looking at the *justification* zone, we find complex zone-feature relations to subsume the courts judgement. There is a common use of connectives in certain participle constructions to argue for or against an aspect of the subject.

4.3. Other Zone-Feature Relations

Beside linguistically more relevant features like keywords and grammatical categories in particular, there are also features which are non-linguistic, including layout and special patterns. Layout features that can be found in the corpus are, for example, significant linebreaks used to part sections in text. Usually, they appear in global or complex zone-feature relations. Even though they are not linguistic at all, they are helpful to identify content zone borders.
More related to keywords are special patterns covering dates and file identifier used by the court as well as legal paragraphs. Such strings can be described by regular expressions. They have the same properties like keyword-features in general, therefore can be sufficient for a content zone.
A summary of all zones described in this sections is shown in table 2.

Finally, there are local zones that need information beyond keyword matching to be identified probably. These are, for example, consequences both for plaintiff and defendant, or also the the views of plaintiff and defendant, which use subjunctive to be discriminated from zones that state the view of the court.

5. A Description Language for Linguistic Features

The observation that zone-feature relations appear frequently and in relatively stable manner seemed promising enough to define a description language that is able to specify linguistic aspects of content zones. Integration to parsing and retrieval frameworks can enhance operations on documents. Though we are restricting research to the legal domain, we also aimed at modeling a more flexible language that does not suffer from text genre specific restrictions when transformed to documents of another domain.

Minimal zfr	**features**
court	court type and town name
date	date of judgement
file	file identifier
formula	keyword phrase
Complex zfr	**features**
judgement	special subordinate clause, conjunctions, keywords
sec. judgement	keyword set
plaintiff/defendant's view	keywords, subjunctive
plaintiff/defendant's application	keywords, special subordinate clause
subsumtion	conjunctions, subjunctions, adverbs, keywords
Layout zfr	**features**
section (global) and aspect (local) border	linebreaks

Table 2: Zone feature relations.

In our *zone description language* (*zdl*), every text genre specification is stored in an XML-file whose root element is `genre`. At the moment, every `genre` only uses the attribute `name`. By using the element `genre`, it is possible to define a sequence of genre definitions in a single file.
Note that *zdl* does not prescribe any structural constrains to a document of certain genre, for the document validation of the surface structure is already accomplished by the schema we described in section 3..
To simplify the language structure, we use one `zone`-tag to deal with global and local zones. To assign hierarchical dependencies between local and global zones, a special attribute `depID`, which refers to the unique `zoneID` of a superordinate zone, is introduced.
A zone then includes one or more `feature`-elements. To model global zone elements that merely function as container for subordinate zones, we kept `feature`-elememts in zones optional. These `features` apply several attributes for functional description (see table 5.2.).

5.1. Feature Classes and Types

By now, there are three general classes of features included. They are defined by the attribute `type` and a value that reads a specific type of feature. The three cover the aspects of *layout*, *lexis* and *syntax*. Each consists of significant properties that have been observed in the previous section.

Attribute	Definition
zoneID	A unique integer.
name	String that should be self-explanatory
depID	optional reference to superordinate.

Table 3: Attributes of element zone.

Layout features. In our corpus, *layout features* are, for example, significant linebreaks used to part sections in text. Usually, they appear in complex zone-feature relations. Even though they are not linguistic at all, they are helpful to identify content zone borders. Also header phrases can be regarded as layout feature. However, they also have lexical character since they consist of important keywords. For the purpose of this content zone specification, headers are not regarded as layout features.
For the purpose of this paper, layout features can be linebreaks which states a text portion is separated by linebreaks above and below.

Lexical Features. Most of the features found in court decisions are of *lexical* type. Very often, certain keywords like legal terms, fixed wordings and headers (see above) or varying phrases like town or person names are used. In this context, fixed wordings and also certain abbreviations are treated as keyword patterns.
Lexical features are the attribute values of keyword for a single term or phrase for a string of several terms.

Syntactic Features. As shown in section 4.2., syntactic features are more complex than others in general. At this moment, sentence mode and tense are included as well as active and passive. They are specified via indicative, subjunctive, tense-present tense-past tense-perfect tense-pastperfect, active and passive.

5.2. Feature Attributes

For every feature element, a number of attributes is used. They are needed to

- name them (name),
- give them a unique identifier for reference purpose (featureID),
- make them comparable via frequency (weight),
- tell that a feature is sufficient for zone identification (sufficiency),
- and to assign a specific parameter to a certain processing module (e.g. a tagger) for parsing purpose which URI is defined in moduleURI.

The attributes sufficiency and parameter are optional while all others are obligatory. Every feature is defined just once in zdl because every feature gets a unique identifier. If, for example, a zone can have a number of optional synonymous keywords, each has to be declared as a single feature.

Attribute	Definition
featureID	A unique integer.
name	String that should be self-explanatory.
type	Classification of feature type.
sufficiency	Binary value for sufficient prerequisites.
weight	Frequency dependent weight for this feature.
parameter	A specific value passed on to the expert module.
moduleURI	Adresses the independent expert module.

Table 4: Attributes of element feature.

5.3. Examples of ZDL-Definitions

In the following example, several aspects of zone-feature relation definition are illustrated.

```
<zone zoneID="23" name="defendant-view" depID
    ="18">
<feature featureID="39" name="defendant-view-mode
    " type="subjunctive" sufficiency="true"
    weight="0.5" parameter="subjunctive"
    moduleURI="www.ling.uni-potsdam.de"/>

</zone>
```

The listing above shows a feature in a zone. First, the zone is defined by a unique ID (23) and a name (defendant-view). Then the reference to the superordinate zone's zoneID is stated (18). This is the ID of the facts zone. Now the feature tag is set and an ID as well as a name analogous to the zone above are declared. The name always extends the name of the zone. After that, the type of this feature (subjunctive) ist set, and the fact that the existence of this feature will maximize the probability of the zone (sufficiency=true). Following the frequency weight real number, a parameter value subjunctive is stated and passed to the expert module via its URI. In this case, just a simple URL was posted to illustrate the functioning.

The weight of a feature is determined by its frequency in the zone in the training corpus. If a feature's presence is sufficient, its weight is always maximized. In a content zone parsing process, the weight will determine the probability of a certain zone label for a text segment.

6. Conclusion and Outlook

The language introduced is able to represent linguistic knowledge in German court decision documents according to the content zone concept presented above. However, at the moment, it just sketches the complex nature of coherence and argumentation.

While trying to cover significant text aspects of German court decision documents, we tried to keep the description language flexible, so conversion to another legal related text genre or a completely different one is possible.

The zone description language is planned to define a specification which will be used by a parsing framework for content zone analysis of German court decisions. The framework is planned to manage a collection of parsing and evaluation sub-processes to classifiy the content of a zone. Accordingly, segments will be labelled for further processing. The linguistic knowledge gained in this process can further be used to enhance retrieval queries as well as text summarization tasks in the domain of German court decisions.

7. References

Mariangela Biasiotti, Enrico Francesconi, Monica Palmirani, Giovanni Sartor, and Fabio Vitali. 2008. Legal informatics and management of legislative documents. Technical report, Global Centre for ICT in Parliament Working Paper No. 2.

Heike Bieler, Stefanie Dipper, and Manfred Stede. 2007. Identifying formal and functional zones in film reviews. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*. Antwerpen.

Dietrich Busse. 2000. Textlinguistik und rechtswissenschaft. In Gerd Antos et al., editor, *Text und Gespra"chslinguistik. Ein internationales Handbuch*.

Jan Engberg. 1992. Signalfunktion und kodierungsgrad von sprachlichen merkmalen in gerichtsurteilen. *Hermes - Journal of Language and Communication Studies*.

Ben Hachey and Claire Grover. 2006. Extractive summarization of legal texts. *Artificial Intelligence and Law*, 14:305–345.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: The detection, classification and structure of arguments in text. In *Proceedings of the ICAIL 2009*. Barcelona, Spain.

Marie-Francine Moens, Caroline Uyttendaele, and Jos Dumortier. 1996. Abstracting of legal cases: The salomon experience. In *Proc. of the 6th International Conference on Artificial Intelligence and Law*. Melbourne.

Erich Schweighofer. 1999. The revolution in legal information retrieval or: The empire strikes back. *Journal of Information, Law and Technology*.

Manfred Stede and Florian Kuhn. 2009a. Document structure and argumentation in german court decisions. In *ICAIL 2009 NaLEA Workshop*.

Manfred Stede and Florian Kuhn. 2009b. Identifying the content zones of german court decisions. In *2nd LIT 2009 Proceedings*.

John M. Swales. 1990. *Genre Analysis: English in Academic and Research Settings*. Cambridge.

Simone Teufel and Marc Moens. 2002. Summarizing scientific articles – experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4).

Stefan Walter and Manfred Pinkal. 2005. Linguistic support for legal ontology construction. In *Proceedings of ICAIL*.

Stephan Walter. 2008. Linguistic description and automatic extraction of definitions fro german court decisions. In *Proceedings of the LREC 2008*.

D.C. Walton, C. Reed, and F. Macagno. 2008. *Argumentation Schemes*. Cambridge University Press.

Controlling the language of statutes and regulations for semantic processing

Stefan Hoefler and Alexandra Bünzli

University of Zurich, Institute of Computational Linguistics
Binzmühlestrasse 14, 8050 Zurich, Switzerland
hoefler@cl.uzh.ch, buenzli@cl.uzh.ch

Abstract

Controlled Legal German (CLG) is a subset of legal German specifically designed to facilitate the semantic processing of Swiss statutes and regulations. In this paper, we describe the strategies CLG employs to reduce ambiguity and underspecification in such texts, and the methods it uses to maintain proximity to conventional legal language. The presented discussion suggests that, if existing synergies are properly exploited, the concept of controlled natural language can be of benefit to the semantic processing of legal texts as well as to legislative drafting.

1. Introduction

The last two decades have brought substantial progress in the development of formal logical representations of legal knowledge and of methods to perform automated legal reasoning with these representations (Rissland et al., 2003). However, as McCarty (2007, p. 217) observes,

> [o]ne of the main obstacles to progress in the field of artificial intelligence and law is the natural language barrier. Since the raw materials of the law are embodied in natural language – cases, statutes, regulations, etc. – the designer of a knowledge-based legal information system today must translate them, by hand, into a formal language, just to get started.

Since a manual translation of legal texts into formal logical representations is both time-consuming and error-prone, the employment of natural language processing techniques seems to be the only viable option to bridge the gap between legal texts and knowledge-based legal information systems. While state-of-the-art methods of natural language processing have come to deliver fairly decent results (McCarty, 2007), they continue to struggle with the notoriously difficult resolution of natural language ambiguity and underspecification.

The Collegis project (Controlled Language for Legal Information Systems) addresses this problem from the perspective of legislative drafting. We develop Controlled Legal German (CLG), a restricted version of Swiss legal German specifically designed to facilitate the semantic processing of statutes and regulations.

Controlled languages restrict the vocabulary, syntax and/or semantics of a natural language in order to reduce its ambiguity and complexity. While early versions of controlled languages were mainly devised to improve the readability and translatability of texts, recently, the method has been used to define subsets of natural languages that can be unambiguously translated into formal logic (Pool, 2006; Fuchs et al., 2008). Controlled languages have been developed for the domains of technical documentation and requirements engineering and for general-purpose knowledge representation. There have also been first attempts to apply the method to defining business rules (Spreeuwenberg and Anderson Healy, 2009) and writing contracts (Pace and Rosner, 2009).

In this paper, we build on a proposal by Hoey and Walter (1988) and introduce legislative drafting as another promising area of application. Legislative drafting, by definition, already exerts a certain degree of control on legal language, thereby pursuing aims similar to those of controlled languages: the reduction of ambiguity and sufficient specification of rules. While there have been studies on improving the understandability of legal language (Wydick, 2005; Neumann, 2009), no controlled legal language has as yet been developed for the purpose of facilitating automated semantic processing.

The remainder of the paper is organized as follows. We first give an overview of the rationale behind CLG and the methods it applies. After detailing the aims of CLG (section 2), we introduce the methods it uses (section 3) and illustrate with a specific example how these methods are applied (section 4). Afterwards, we demonstrate how CLG exploits conventions that already exist in Swiss legal language (section 5) and discuss approaches to controlling underspecification in statutes and regulations (section 6). After describing the current state of development of CLG (section 7), we conclude with the presentation of a brief proposal for the evaluation of controlled legal languages (section 8) and with a discussion of the potential and limitations of the approach for both semantic processing of legal texts and legislative drafting (section 9).

2. Aim

The goal we pursue with the development of CLG is to provide a language for Swiss statutes and regulations whose semantics can be understood by humans and processed by computers. To allow for an automatic translation of such texts into formal logical representations which can e.g. be fed to some automated inference system, CLG aims at reducing natural language ambiguity. However, while CLG eliminates lexical ambiguity in function words and law-specific expressions, it leaves the interpretation of content words to the terminology databases and ontologies of its users. CLG does therefore not infringe on the often intended vagueness and open-textured nature of the concepts represented by content words (Gardner, 1987). Like other

controlled languages that aim at providing an interface to some sort of formal logic, such as ACE (Fuchs et al., 2008) or PENG (Schwitter and Tilbrook, 2006), CLG is mainly concerned with the reduction of syntactic and semantic ambiguity.

Syntactic ambiguity occurs in situations where a sentence can be assigned more than one syntactic structure. Typical examples are so-called attachment ambiguities: in sentence (1), the prepositional phrase *im Bereich der Logistik* ('in the sector of logistics') could theoretically be attached to *deckt* ('supplies'), to *Bedarf* ('need'), to *Güter und Dienstleistungen* ('goods and services'), or only to *Dienstleistungen* ('services').

(1) Das Bundesgericht deckt seinen Bedarf an Gütern und Dienstleistungen im Bereich der Logistik selbständig. (Art. 25a Abs. 2 BGG[1])

'The Federal Supreme Court supplies its need for goods and services in the sector of logistics autonomously.'

Semantic ambiguity, on the other hand, occurs if a sentence has only one syntactic structure but can be assigned two or more non-equivalent logical representations. A typical example is scope ambiguity. Without contextual knowledge, it cannot be determined whether sentence (2) requires that each representative (or rather the representatives of each party) show a separate letter of attorney, or whether it means that they should provide one letter of attorney together.

(2) Die Parteivertreter und -vertreterinnen haben sich durch eine Vollmacht auszuweisen. (Art. 40 Abs. 2 BGG)

'The party representatives have to identify themselves with a letter of attorney. '

Besides reducing ambiguity, CLG aims at preventing types of underspecification that warrant unintended inferences. This goal is described in more detail in section 6.

3. Methods

Generally, controlled natural languages use the following methods to reduce natural language ambiguity and complexity:

- *Construction rules*
  Construction rules restrict the number of words and constructions that can be used, thus prohibiting the use of specific ambiguous words and constructions.
- *Interpretation rules*
  Interpretation rules assign default interpretations to the remaining ambiguous words and constructions.
- *Paraphrases*
  Paraphrases suggest alternative ways of expressing the respective other meaning of an originally ambiguous word or construction.

[1]Bundesgerichtsgesetz (Federal Supreme Court Act), SR 173.100

One of the main problems of the method of controlled natural language is the fact that there is a trade-off between the level of control a language exhibits (and thus its processability) and its expressiveness, naturalness and user-friendliness. Most existing controlled natural languages, especially those aspiring to provide an interface to some kind of formal logic, consequently have only very limited expressiveness, and several of them include constructions which border on naturalness at best (Pool, 2006). CLG differs from these languages as it needs to be expressive enough to render the contents of statutes and regulations, and natural enough to be understood and accepted by non-expert human readers. We employ three methods to maximize CLG's proximity to ordinary legal language:

- *Syntactic sugar*
  As the naturalness of specific control mechanisms may vary from context to context, CLG usually provides more than one way of controlling an individual phenomenon.
- *Variable-depth control*
  For certain phenomena, CLG provides multiple levels of control, which can be switched on or off by the user, depending on the requirements of the target application.

  If further specificity is not required, certain ambiguous constructions are only assigned underspecified logical representations. The treatment of such constructions is then left to the tools that process the logical representations.[2]
- *Interactive control*
  Some phenomena are not controlled statically but resolved dynamically by providing an authoring tool that asks the user to specify the intended meaning upon each occurrence of the respective ambiguous construction (Macias and Pulman, 1995).

  To guarantee transparency, the choices made by the user are recorded in a so-called disambiguation protocol, which is to be stored together with the text.

In the next section, we illustrate with a specific example how these methods are applied.

4. Applying the methods

In Swiss statutes and regulations, indefinite noun phrases in subject position usually indicate what the respective norm is about, i.e. they introduce the "subject matter" of the norm (Caussignac et al., 2000). However, the indefinite article (*ein/eine/ein* 'a(n)' in singular; ∅ in plural) is ambiguous at this position: it can have an existential interpretation, as in example (3), or a generic interpretation, as in example (4), which can be represented as universal quantification (Gamut, 1991; Cohen, 2001).[3]

[2]Attempto Controlled English (Fuchs et al., 2008) uses this method for plural ambiguities and copula; Computer Processable Language (Clark et al., 2005) employs it for PP-attachment ambiguities.

[3]In statutes and regulations, the generic interpretation of the indefinite article does not express the prototypical features of a

(3) Ein Mitglied der Universitätsleitung führt den Vorsitz. (§ 67 Abs. 2 UniO UZH[4])

'A member of the Executive Board of the University acts as chair.'

$\exists x : member(x) \wedge ...$

(4) Ein Titel [...] kann von der Erweiterten Universitätsleitung auf Antrag der Fakultät entzogen werden, wenn die Inhaberin oder der Inhaber die Interessen der Universität ernsthaft verletzt. (§ 8 Abs. 7 UniO UZH)

'A title [...] can be revoked by the Extended Executive Board of the University at the request of the faculty if the holder seriously violates the interests of the university.'

$\forall x : title(x) \rightarrow ...$

One way to control this ambiguity is to define a construction rule that prohibits the use of the indefinite article *ein* altogether and to offer paraphrases for its two interpretations in order to maintain the expressiveness of the language. An existentially quantified subject matter could be introduced by *mindestens ein* ('at least one') or *genau ein* ('exactly one'); for a universally quantified subject matter, one could use the determiner *jeder* ('every'). The problem with this solution is that it represents a significant deviation from conventional legal language, where the use of *ein* is very common while the use of *mindestens/genau ein* and *jeder* is rare and more marked. Adopting this construction rule would thus substantially decrease the naturalness of CLG.
The solution at the other end of the scale is to resolve the ambiguity caused by the use of *ein* interactively, i.e. to devise an authoring system that asks the user to indicate for every occurrence of a subject matter introduced by *ein* whether existential or universal quantification is intended. However, while interactive control is a viable option for relatively rare phenomena, it is clearly not user-friendly for phenomena that occur as frequently as the indefinite article. This solution too must be rejected.
The only remaining method of control is the definition of an interpretation rule that identifies one of the two readings of *ein* as the default interpretation. In statutes and regulations, indefinite plural noun phrases in subject position generally exhibit a generic reading and are thus to be represented as universally quantified. Sentence (5) provides an example.

(5) Dienstleistungen sind in der Regel mindestens kostendeckend in Rechnung zu stellen. (§ 3 Abs. 3 UniO UZH)

'Services usually have to be charged so that at least the costs covered.'

$\forall x : service(x) \rightarrow ...$

kind but states a rule that applies to every instance of that kind. – This observation can be conceived as a CLG interpretation rule defining that the generic reading of the indefinite article is interpreted as universal quantification.

[4]Universitätsordnung der Universität Zürich (University Regulation of the University of Zurich), SR 415.111

As CLG aims at staying close to conventional legal language, it would make little sense to define the existential reading as the default interpretation of indefinite plural noun phrases. For indefinite singular noun phrases, neither interpretation can be considered conventional. To keep the number of rules that users of CLG need to master low, we apply one and the same interpretation rule to both the singular and the plural version of the indefinite article: indefinite noun phrases are interpreted as universally quantified in subject position (and as existentially quantified elsewhere; see section 7).
The definition of such an interpretation rule entails that example (3) needs to re-phrased to obtain existential quantification. Two options are available. The first is to make the existential quantification explicit by using determiners such as *mindestens ein* ('at least one') or *genau ein* ('exactly one'). In the present example, however, these determiners do not sound particularly natural and potentially confuse the reader as they seem to be marked pragmatically:

(6) Genau ein Mitglied der Universitätsleitung führt den Vorsitz.

'Exactly one member of the Executive Board of the University acts as chair.'

Alternatively, the noun phrase *ein Mitglied der Universitätsleitung* can be moved away from the subject position. This effect can be achieved by using a passive construction such as (7). For the present example, this second solution provides a sentence that both feels natural and is interpreted in the intended way in CLG.

(7) Die Forschungskommission wird von einem Mitglied der Universitätsleitung präsidiert.

'The research committee is chaired by a member of the Executive Board of the University.'

Sentence (7) is preferable to (3) not just from the perspective of semantic processing but also from the perspective of legislative drafting. First, the subject of a norm should usually indicate what this norm is about. The present norm is not about some member of the Executive Board of the University but about the research committee. Second, the rephrased version indicates explicitly what the chair is *of* (namely the research committee); in the original version, this information has to be inferred from the context. We come back to such phenomena of underspecification in section 6.
Depending on their target application, some users of CLG may not want to commit to the aforementioned interpretation rules. Answer extraction, for instance, can cope without the explicit specification of quantification (Mollá, 2001). As CLG pursues a policy of variable-depth control, it therefore also provides the option of leaving the quantification of indefinite noun phrases underspecified. In that case, the aforementioned interpretation rules do not apply.

5. Exploiting domain-specific conventions

Since its aims are similar to those of controlled natural language, conventional legal language itself provides mechanisms to control certain types of ambiguity. Whenever possible, CLG exploits these already existing mechanisms.

CLG makes, for instance, use of the fact that some words and constructions that are ambiguous in full natural language have acquired a default interpretation in legal language. In ordinary German, the adverb *grundsätzlich*, modifying an obligation or permission, can have two directly opposed interpretations: if interpreted in the sense of 'strictly' or 'categorically', it denotes that the respective rule does not allow for exceptions; if interpreted as 'generally' or 'in principle', it indicates that the rule allows for exceptions, which is particularly relevant in the context of defeasible reasoning. By convention, *grundsätzlich* is always used in the latter sense in Swiss legal German. CLG therefore devises an interpretation rule defining that *grundsätzlich* is always interpreted as indicating the admissibility of exceptions:

(8) Die Veröffentlichung der Entscheide hat grundsätzlich in anonymisierter Form zu erfolgen. (Art. 27 Abs. 2 BGG)

'*In principle*, the decisions must be published in anonymized form.'

Note that unlike ordinary adverbs, *grundsätzlich* does not modify the verb but the obligation as a whole. CLG defines a number of words and fixed expressions that are not interpreted like other items of the same grammatical category but obtain domain-specific interpretations. Table 1 lists the most common of them.

Another example of a phenomenon for which CLG exploits existing domain-specific methods of control is attachment ambiguity in complex coordination structures. Sentences like (9) are difficult to parse not only for computers but also for humans. It is thus in the best interest of both NLP and legislative drafting to control the attachment ambiguities they contain.

(9) In Fünferbesetzung entscheiden sie ferner über Beschwerden gegen referendumspflichtige kantonale Erlasse und gegen kantonale Entscheide über die Zulässigkeit einer Initiative oder das Erfordernis eines Referendums. (Art. 20 Abs. 3 BGG)

'In a composition of five, they furthermore decide on appeals against cantonal decrees subject to referendum and against cantonal decisions on the admissibility of an initiative or the necessity of a referendum.'

CLG defines an interpretation rule stating that constituents are always attached to the closest possible candidate. While this rule can be easily applied to relatively simple sentences, it is clearly not user-friendly enough, both in terms of writability and readability, for complex coordination structures such as (9). To disambiguate such structures, CLG includes a means provided by conventional legal language: ellipses are removed by repeating all elements in each conjunct (in this case, the phrase *kantonale Entscheide* 'cantonal decisions' is repeated) and the conjuncts are listed in enumerations introduced by letters, as shown in (10).

(10) In Fünferbesetzung entscheiden sie ferner über Beschwerden gegen:

a. referendumspflichtige kantonale Erlasse;
b. kantonale Entscheide über die Zulässigkeit einer Initiative;
c. kantonale Entscheide über das Erfordernis eines Referendums.

'In a composition of five, they furthermore decide on appeals against:

a. cantonal decrees subject to referendum;
b. cantonal decisions on the admissibility of an initiative;
c. cantonal decisions on the necessity of a referendum.'

6. Controlling underspecification

Besides ambiguity, underspecification is the main issue that a controlled legal language needs to address. We can distinguish two types of underspecification in statutes and regulations.

The first type occurs where legislators deliberately refrain from specifying certain details. Sentence (11) may serve as an example.

(11) Die Bundesversammlung wählt die Richter und Richterinnen. (Art. 5 Abs. 1 BGG)

'The Federal Assembly elects the judges.'

In general, plural noun phrases can have a distributive reading (each judge is elected individually) and a collective reading (the judges are elected as a body).[5] As the distributive interpretation is far more frequent in statutes and regulations, CLG defines it as the default interpretation. To express the collective reading, a singular term has to be used (e.g. *das Gericht* 'the court'). This strategy can also be frequently found in existing legal texts. However, even with such an interpretation rule being applied, sentence (11) remains indeterminate: it does not specify the exact conditions under which an individual judge is considered elected. Even if the Federal Assembly elected the judges as a body, each judge might considered elected individually by this act. The legislator deliberately leaves the conditions under which a judge needs to be elected undetermined here; CLG reflects this fact despite the application of an interpretation rule.

The second type of underspecification poses a much more substantial problem to the semantic processing of statutes and regulations. It occurs in passages that warrant unintended inferences if they are not further specified and that are therefore potentially harmful to correct automated reasoning. Sentence (12) is an example.

(12) Bei der Geburt eines Kindes hat der Angestellte Anspruch auf eine einmalige Zulage von 530 Franken. (Art. 55 Abs. 1 AngO ETH-Bereich[6])

'Upon the birth of a child, the employee is entitled to a one-time allowance of 530 francs.'

[5] The treatment of plural ambiguities in controlled language is thoroughly discussed in Schwertel (2000).

[6] Angestelltenordnung ETH-Bereich (Employee Regulation ETH-Domain), SR 172.221.106.2

Word / Expression	Translation	Function
muss/müssen, hat/haben zu*	must, have to	marks a rule as an obligation
darf/dürfen, kann/können*	may, can	marks a rule as a permission
grundsätzlich, in der Regel*	in principle	indicates that a rules allows for exceptions
gemäss, im Rahmen (von)	according to, within the scope (of)	indicates the applicability of another rule
(gilt) sinngemäss	(applies) analogously	indicates that a rule is applicable in adapted form
insbesondere, namentlich*	in particular	indicates that a specific case is made explicit
bei (+NP)	upon (+NP), ∼if	condition in the form of a PP; cf. example (12)

Table 1: Examples function words and fixed expressions with conventionalized domain-specific interpretations. (Expressions marked with an asterisk are contained in CLG 1.0; cf. section 7.)

The problem sentence (12) poses for semantic processing is that the condition (*at the birth of a child*) apparently does not share any discourse referent with the consequence (*the employee is entitled to a one-time allowance of 530 francs*). The sentence does not specify explicitly that the employee does not receive an allowance on the occasion of the birth of just *any* child but only if he or she is the parent of that child. Human readers will easily infer this missing bit of information from the context and thus reduce the number of warranted inferences. An automated reasoner, on the other hand, may in the worst case combine the logical representation of (12) with the fact that approximately 216,000 children are born every day, and deduce that an employee is to receive total allowances of 114,480,000 francs per day.
To avoid this problem, a controlled legal language may prescribe that the condition of a conditional norm always has to share a discourse referent with its consequence. This requirement can be fulfilled, for instance, by augmenting the condition with a relative clause:

(13) Bei der Geburt eines Kindes, *gegenüber dem er elterliche Pflichten hat*, hat der Angestellte Anspruch auf eine einmalige Zulage von 530 Franken.

'Upon the birth of a child *toward whom he or she has parental duties*, the employee is entitled to a one-time allowance of 530 francs.'

The same effect is achieved if another condition is added at the end of the sentence:

(14) Bei der Geburt eines Kindes hat der Angestellte Anspruch auf eine einmalige Zulage von 530 Franken, *sofern er gegenüber dem Kind elterliche Pflichten hat.*

'Upon the birth of a child, the employee is entitled to a one-time allowance of 530 francs, *provided that he or she has parental duties toward the child.*'

Note that the application of this rule is not only beneficial to semantic processing but also to legislative drafting. Had they been forced to provide the additional specification, legislators would have become aware of an overlooked regulatory loophole, namely that biological parents who are not liable for support should not be entitled to an allowance while foster parents should.
An alternative solution to controlling the phenomenon becomes available if one recognizes that the noun *Kind* ('child') is in fact ambiguous: it can denote a young human, or it can denote someone's direct offspring. In the latter sense, *Kind* is a relational noun, whose logical representation takes not one but two arguments: $child\_of(x, y)$. The noun *Kind* is thus implicitly anaphoric, referring to some other entity in the text.[7] The problem is then to constrain the field of potential antecedents so that the implicit reference is unambiguous. The guidelines for legislative drafting provided by the Swiss Confederation (BJ, 2007) and by the Canton of Zurich (ZH, 2005) constrain the use of pronouns – another type of anaphoric references – in statutes and regulations: pronouns may only refer to entities within the same article and they may only refer to either the subject of the main clause or to the subject of the immediately preceding sentence. The same rule can now be applied to the implicit anaphoric references created by relational nouns: their use may be constrained to referring to either the subject of the main clause (as is the case in our example) or, if they are part of that subject themselves, to the subject of the immediately preceding sentence – provided that that sentence is in the same article as the sentence containing the relational noun. It needs to be said, however, that this rule is not yet implemented in CLG 1.0, the version of the language representing the current state of development, which we will briefly describe in the next section.

7. State of development

The state of development of Controlled Legal German is reflected in version 1.0 of the language, which is documented in Hoefler and Bünzli (2010). CLG 1.0 provides the basic syntactic and semantic inventory to express simple norms (obligations, permissions, prohibitions; including norms stating duties and responsibilities) as well as legal definitions. Example (15) provides a typical CLG 1.0 sentence and the logical representation it maps onto.

(15) Radfahrer müssen mindestens zwei rote Rückstrahler tragen, sofern sie keine Ausnahmebewilligung haben.

'Cyclists must wear at least two red reflectors, unless they have (if they do not have) a certificate of exception.'

$$\begin{array}{lll} \mathcal{O} & \forall x: & [radfahrer(x) \wedge \\ & & \neg\exists y : [ausnahmebewilligung(y) \wedge \\ & & \exists e : has(e, x, y)] \\ & \rightarrow & \exists^{\geq 2} z : [roter\_rueckstrahler(z) \wedge \\ & & \exists f : traegt(f, x, z)]] \end{array}$$

[7] We thank an anonymous reviewer for pointing this out.

Sentence (15) illustrates the following characteristics of the formal semantics underlying version 1.0 of Controlled Legal German. CLG 1.0 can be unambiguously mapped onto predicate logic representations that are augmented with deontic operators for obligation ($\mathcal{O}$), permission ($\mathcal{P}$) and prohibitions ($\neg\mathcal{P}$). Since content words are translated into predicate symbols, the potential open-texturedness of the concepts they represent is preserved. The events and states represented by verbs are reified (and quantified). Adjectives used in an attributive manner and the nouns they modify are, for now, contracted into a single logical predicate. CLG 1.0 includes means to express existential and universal quantification, as well as counting quantifiers. It does, however, not include elements of temporal and intensional logics. These concepts are planned to be added to the standard in CLG versions 2.x and 3.x respectively.
The following list provides an overview of the range of syntactic constructions that are available in CLG 1.0:

1. Only present tense are permitted.
2. Sentences have canonical word order (the subject preceding objects and adverbials).
3. Both active and passive voice is permitted.
4. Nouns can currently be modified by (a) adjectives, (b) participle constructions, (c) relative clauses, but not by prepositional phrases (with the exception of the prepositional phrase denoting the agent of a nominalized verb).
5. Verbs can be modified by (a) adverbs, (b) prepositional phrases.
6. Main clauses may contain a modal verb; main clauses without modal verb are assumed to be obligations.
7. Nouns, verbs and adjectives can be coordinated; coordinations may be put in the form of enumerations (cf. section 5).
8. Attributes in genitive case are only permitted to express the direct object of nominalized verbs or the complements of relational nouns.
9. Conditional clauses and relative clauses are the only permissible subordinate clauses.
10. Complement clauses and adverbial clauses are not permitted (with the exception of conditional clauses).
11. There are special formulaic expressions to list duties and responsibilities, such as e.g. *X hat die folgenden Aufgaben und Kompetenzen* ('X has the following duties and responsibilities').

The semantics of CLG 1.0 sentences is controlled by the following seven interpretation rules:

1. Modal verbs have wide scope over the rest of the sentence.
2. Subjects have wide scope over Objects and Adverbials.
3. Pronouns refer to the subject of the sentence or, if they are part of the subject, to the subject of the immediately preceding sentence.
4. Indefinite noun phrases are interpreted generically if they are the subject of a sentence and existentially elsewhere.
5. Plurals are interpreted distributively. If a collective reading is intended, a singular term has to be used.
6. Definite noun phrases presuppose existence and uniqueness and are interpreted referentially. Definite plurals are interpreted distributively and universally.
7. Attachment ambiguity is resolved by attaching the constituent in question to the closest potential antecedent; if that antecedent is a conjunct, the constituent is attached to the whole coordination.

For a detailed account of CLG 1.0, we refer to Hoefler and Bünzli (2010). Evidently, CLG 1.0 is not yet expressive enough to be used in legislative drafting. It can, however, be employed to model simple norms in a way that provides a formal specification and is yet understandable by non-expert human readers.

8. Proposal for evaluation

As the development of CLG is still work in progress, a thorough evaluation of the controlled natural language can not yet be provided. In any case, before such an evaluation can be undertaken, one needs to define how a controlled language that aims at facilitating the semantic processing of statutes and regulations is to be assessed in the first place. We propose that such a controlled language has to be evaluated for the following criteria:

- *Expressiveness*
  An ideal controlled legal language should be able to express all propositions that conventional legal language can express. The expressiveness of a controlled legal language can be assessed by determining what percentage of the content of a chosen statute or regulation (e.g. how many of the individual norms contained in that text) it can express.
- *Proximity to conventional legal language*
  An ideal controlled legal language should be indistinguishable from conventional legal language in terms of style. As a first approximation, the degree to which a controlled legal language covers conventional legal language can be evaluated by assessing how many articles of a chosen statute or regulation need to be altered if that text is to be translated into the controlled language. If only few passages have to be altered, the respective controlled language can be considered stylistically close to conventional legal language. However, the need for rephrasing does not necessarily imply that the resulting text deviates from the conventions of legal language. It may still be perfectly acceptable. The stylistic acceptability of substantially altered texts therefore requires additional assessment by human legal editors.

As we have already pointed out above, the degree to which a controlled language covers conventional legal language can be maximized by the use of syntactic sugar, the employment of control mechanisms that reflect the frequency distributions in the reference texts, and the provision of authoring systems that allow for certain phenomena to be disambiguated interactively.

Both criteria, expressiveness as well as proximity to conventional legal language, can only be assessed in a controlled legal language if a corpus of semantically analyzed reference texts is available. We are currently building such a corpus, starting with the Federal Supreme Court Act and the Regulation of the University of Zurich, from which we quoted in this paper.

9. Discussion and conclusion

The conditions encountered in legislative drafting seem ideal for an application of controlled natural language. Statutes and regulations are written in a highly conventionalized language that contains restrictions aimed at preventing ambiguity. Due to this shared goal, the properties of legal language are not unlike those of typical controlled natural languages. Controlled Legal German uses these synergies to facilitate the drafting of statutes and regulations that can be automatically translated into formal logical representations. It thus attempts to bridge the gap between legal texts, written in natural language, and knowledge-based legal information systems, operating with formal logical representations.

In this paper, we have presented the general rationale behind CLG and introduced the methods it applies to prevent ambiguities and underspecification. We have shown that CLG consists of a set of recommendations in the form of construction and interpretation rules of variable depth, accompanied by suggested paraphrases and options for interactive ambiguity resolution. These recommendations explain how statutes and regulations can be formulated in a way that enables automatic semantic processing. In parallel to defining such rules, we are working on combining them into a comprehensive formal description of a subset of Swiss legal German that can be translated deterministically into formal logical representations.

At this point, some remarks on the limitations of the approach of applying controlled natural language to legal texts are in place. A first limitation of the approach pertains to the availability of adequate logical representations for the content expressed linguistically in norms. Not all linguistic phenomena can easily be represented in formal logic: temporal relations or intensional contexts, for instance, already require a rather complex machinery of operators and axioms. But even apparently simple linguistic constructions such as attributive genitives or opaque adjectives do not have straightforward logical representations. There will always be some phenomena that have to be treated as "black boxes" or modeled in a grossly simplified manner. Any logical representation derived from a norm written in controlled natural language will thus only capture the content of that norm to a certain degree of granularity.

A second limitation of the approach refers to the fact that controlled natural languages such as CLG, ACE or PENG may be able to reduce (or, in the ideal case, eliminate) certain types of ambiguity but cannot (and do not aim to) remove vagueness. The predicates of any logical representation will still stand for concepts whose definition may be vague or open-textured. On the one hand, this fact reflects a reality of legal language, where vagueness and indeterminacy is often positively intended (Nussbaumer, 2005). On the other hand, it means that being able to derive a logical representation from a legal text written in a controlled language does not entail that one will automatically be able to perform meaningful legal reasoning over such a representation. While certain inferences can be drawn purely on the basis of the logical representations of a statute or regulation by treating the logical predicates and the potentially vague concepts they stand for as black boxes, deeper automated reasoning will in addition at least require extensive ontologies modeling world knowledge.

A third limitation pertains to the controlled natural language itself. It is to be expected that extensive control of ambiguity will lead to a certain reduction of the expressiveness of legal language. The future development of CLG will have to show whether this reduction can be kept at a level at which it does not seriously impede the usability of CLG for legislative drafting. While experience shows that many types of ambiguity can be controlled by the methods described, underspecification will continue to pose a problem. It will most likely not be possible for a controlled language to prevent the vast number of situations in which a human writer may underspecify some of the information required for accurate reasoning.

Finally, the success of a controlled legal language will depend on its acceptance by professional legal editors: CLG must be easy to learn and close to conventional legal language both in terms of the propositions it can express and the stylistic means it provides. It is too early to speculate if it will be possible to develop a controlled version of legal German that is accepted by its potential users. Using CLG for didactic purposes and for the conceptualization of norms rather than for actual legislative drafting may be a fallback strategy. However, there are three factors that at least have the potential to exert a positive influence on the acceptability of employing controlled natural language in legislative drafting. First, professional legal editors are domain-specialists that are used to (and well capable of) following linguistic guidelines. The application of such rules may be additionally supported by specifically designed authoring tools (Schwitter et al., 2003). Second, there is some chance that we will be able to show that the employment of a controlled legal language can also be beneficial to legislative drafting itself. In this paper, we have briefly demonstrated how CLG can help legal editors detect regulatory loopholes they would otherwise have overlooked. Eventually, however, the acceptance of controlling legal language for semantic processing will be served best if it grants access to AI & Law applications of evident practical use beyond legislative drafting.

Acknowledgements. The authors wish to thank the Central Language Services of the Swiss Federal Chancellery for

their advice on the conventions of legal language and the process of legislative drafting, four anonymous reviewers for their valuable comments and suggestions, and Michael Hess for his scientific and institutional support of the Collegis project.

10. References

BJ (Bundesamt für Justiz), Bern, 2007. *Gesetzgebungsleitfaden: Leitfaden für die Ausarbeitung von Erlassen des Bundes*, 3. edition.

G. Caussignac, C. Eberhard, P. Häusler, D. Kettiger, D. Pulitano, and R. Schneider, 2000. *Rechtsetzungsrichtlinien des Kantons Bern, Modul 4: Sprache*. Justiz-, Gemeinde- und Kirchendirektion und Staatskanzlei des Kantons Bern, Bern.

P. Clark, P. Harrison, T. Jenkins, J. Thompson, and R.H. Wojcik. 2005. Acquiring and using world knowledge using a restricted subset of English. In *FLAIRS 2005*, pages 506–511.

B. Cohen. 2001. On the generic use of indefinite singulars. *Journal of Semantics*, 18(3):183–209.

N.E. Fuchs, K. Kaljurand, and T. Kuhn. 2008. Attempto Controlled English for knowledge representation. In C. Baroglio, P.A. Bonatti, J. Maluszynski, M. Marchiori, A. Polleres, and S. Schaffert, editors, *Reasoning Web: 4th International Summer School 2008*, pages 104–124, Berlin. Springer.

L. T. F. Gamut. 1991. *Logic, Language, and Meaning*, volume 1. The University of Chicago Press, Chicago.

A. Gardner. 1987. *An Artificial Intelligence Approach to Legal Reasoning*. MIT Press, Cambridge, MA.

S. Hoefler and A. Bünzli. 2010. Controlled Legal German 1.0: Einführung und Spezifikation. Technical report, Department of Informatics, University of Zurich, Zurich.

M. Hoey and C. Walter. 1988. Natural language interfaces. In C. Walter, editor, *Computer Power and Legal Language*, pages 135–142, New York. Quorum.

B. Macias and S.G. Pulman. 1995. A method for controlling the production of specifications in natural language. *The Computer Journal*, 38(4):310–318.

L. T. McCarty. 2007. Deep semantic interpretation of legal texts. In *Proceedings of the 11th International Conference on Artificial Intellgigence and Law*, pages 217–224, New York. ACM Press.

D. Mollá. 2001. Ontologically promiscuous flat logical forms for NLP. In *Proceedings of the 4th International Workshop on Computational Semantics (IWCS-4)*, pages 249–265, Tilburg.

S. Neumann. 2009. Improving the comprehensibility of German court decisions. In G. Grewendorf and M. Rathert, editors, *Formal Linguistics and Law*. Mouton de Gruyter, Berlin.

M. Nussbaumer. 2005. Zwischen Rechtsgrundsätzen und Formularsammlung: Gesetze brauchen (gute) Vagheit zum Atmen. In V. K. Bhatia, J. Engberg, M. Gotti, and D. Helier, editors, *Vagueness in Normative Texts*, pages 49–71. Peter Lang, Bern.

G. J. Pace and M. Rosner. 2009. A controlled language for the specification of contracts. In N.E. Fuchs, editor, *Pre-proceedings of the Workshop on Controlled Natural Language (CNL 2009)*. CEUR-WS.

J. Pool. 2006. Can controlled languages scale to the web? In *5th International Workshop on Controlled Language Applications*.

E. L. Rissland, K. D. Ashley, and R. P. Loui. 2003. AI and law: A fruitful synergy. *Artificial Intelligence*, 150(1–2):1–15.

U. Schwertel. 2000. Controlling plural ambiguities in Attempto Controlled English. In *Proceedings of the 3rd International Workshop on Controlled Language Applications*, Seattle, Washington.

R. Schwitter and M. Tilbrook. 2006. Let's talk in description logic via controlled natural language. In *Proceedings of the 3rd International Workshop on Logic and Engineering of Natural Language Semantics*, pages 193–207, Tokyo.

R. Schwitter, A. Ljungberg, and D. Hood. 2003. ECOLE: A look-ahead editor for a controlled language. In *Controlled Translation, Proceedings of EAMT-CLAW03*, pages 141–150, Dublin.

S. Spreeuwenberg and K. Anderson Healy. 2009. SBVR's approach to controlled natural language. In N.E. Fuchs, editor, *Pre-Proceedings of the Workshop on Controlled Natural Language (CNL 2009)*. CEUR-WS.

R. C. Wydick. 2005. *Plain English for Lawyers*. Carolina Academic Press, 5. edition.

ZH (Regierungsrat des Kantons Zürich), Zürich, 2005. *Richtlinien der Rechtsetzung*.

Named entity recognition in the legal domain for ontology population

Mírian Bruckschen[1], Caio Northfleet[2], Douglas da Silva[1], Paulo Bridi[1], Roger Granada[1], Renata Vieira[1], Prasad Rao[2], Tomas Sander[2]

[1]Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Ipiranga Av., 6681. FACIN. CEP 90169-900. Porto Alegre, Brazil.

[2]Hewlett-Packard (HP)
Ipiranga Av., 6681. Building 91B. CEP 91530-000. Porto Alegre, Brazil.

E-mail: mirian.bruckschen@cpph.pucrs.br, caio.northfleet@hp.com, douglas.silva@cpph.pucrs.br, paulobridi@gmail.com, roger.granada@cpph.pucrs.br, renata.vieira@pucrs.br, prasad.rao@hp.com, tomas.sander@hp.com

Abstract

This paper presents the overall problem of privacy risk assessment in the software industry and the difficulty to deal with all normative sources that regulate privacy matters. This problem encompasses the hard task of representing all the relevant information and keep it updated. Ontologies are the main mechanism for domain-specific knowledge representation in the Semantic Web context, but their manual maintenance is expensive and error-prone. Following the ontology learning trend, this paper presents an approach to automatically populate a legal ontology from legal texts through the Named Entity Recognition task and an experiment on this approach. Legal ontologies have been an active topic of research for quite a while, but on specific domains such as data privacy there is still a lack of such resources. The experiment described in this paper is run over a corpus of legal and normative documents for privacy, shows promising results and presents opportunities for the continuation of this research.

1. Introduction

The advent of the Semantic Web has brought the attention of researchers of several areas to applied Artificial Intelligence and Knowledge Representation. Adding semantic features to applications in daily use in the real world has become a goal for many of them, including for those studying AI&Law.

In this context, ontologies are the main mechanism for domain-specific knowledge representation (with logic formality and reasoning as a plus). Legal ontologies have been explored and developed lately (Ajani *et al.*, 2009; Hoekstra *et al.*, 2007), for different purposes and in different subdomains.

Data privacy-specific legislation in ontology form, however, is still a new topic in the context of both semantic technologies and AI&Law research. Most current approaches go in the direction of policy-oriented languages, such as Rei and AIR (Kagal, 2002; Kagal Weitzner & Hanson, 2008). In contrast to this approach, we believe that specialized legal ontologies have a great potential for automating compliance assessment, a matter of great importance for scalable data privacy and accountability scenarios.

Ontologies are the main formalism for expressing domain knowledge in the Semantic Web. They are also complex and require expert skills to be built. Following the ontology learning trend, this paper presents an approach to automatically populate a legal ontology from legal texts through the Named Entity Recognition task, aiming at the discovery of semantic relations at a later time. As an end-goal of this research we wish to provide a resource that can help software industry project managers to calculate, understand and lower privacy risks in their projects.

This paper presents the overall problem of privacy risk assessment in the software industry and the difficulty to deal with all normative sources that regulate privacy matters in Section 2, the developed prototype and its results so far in Section 3, some related work in Section 4 and final remarks in Section 5.

2. Privacy and accountability

The collection and use of personal information from customers is a common practice among companies and governments all around the world. Knowing and applying current privacy legislation and requirements thereby becomes an important requirement for IT projects that might touch personal data and inadequate procedures or data breaches can lead to lawsuits and loss of consumer trust for the company (Mont & Thyne, 2006).

An IT project manager is highly aware and knowledgeable of the business goals an IT project is supposed to achieve. However he will rarely be a privacy expert nor have a privacy expert on his team and so he isn't aware about the legislation that might apply for each context and the actions he may need to take into account to bring a new project into privacy-compliance. Accountable privacy management tries to ensure that each project takes good privacy legislations and best practices into account in such a way that this can also be verified, e.g. by privacy or audit professionals in the company.

Indeed, it is a non-trivial task to determine exactly which rules and requirements apply to a particular project and context. Past efforts have been made on defining a common representation of privacy policies (Kagal, Weitzner & Hanson, 2008; Denker *et al.*, 2003), but we believe that specifications of legislation in ontology can also be useful for recognizing risks concerning penalties and inadequate procedures.

Also, we believe that knowledge representation for legislation is an intermediate step towards semiautomatic compliance verification for systems' specifications according to laws, regulations and best practices. We are confident that this is possible for privacy where we have studied the underlying knowledge domain, but believe that this can potentially be generalized to other compliance concerns.

Even though risk analysis and management are somewhat mature fields (Cybenko, 2006), each scenario is specific and leads to different risks – and takes different measures as well. To measure, understand and fulfill the requirements in order to lower security and privacy risks in a specific scenario, it is necessary to have a broad view of all aspects that involve those risks, all circumstances that lead to it. It is necessary to have a proper and complete representation of the rules that were already written to preserve the customer's information asset. These rules are mostly described in laws, policies and other normative sources.

These normative sources come usually in text form. Each one of these sources is applicable to a specific circumstance: different data destination in a data transfer, different action intended to be done with the data, different type of information (medical and financial records are examples of sensitive information, which should take extra care from companies that deal with it).

Text form is the main way people communicate, but for automatic communication and effective machine processing of any activity, it is necessary to have the interest data in a form that can be read and inferred by computers. In our case, interest data includes the rules that dictate the expected behavior for data management preserving privacy – and how does the projects' designed actions reflect this expected behavior. This can be done automatically by linking actions in the project to laws or policies of the company.

With our research, we intend to achieve a resource to help project managers to deal with privacy issues in their own projects, by linking actions in projects to regulative sources. A first step is automatically identifying these regulative sources. That is what we describe in this paper.

3. Experiments and results

This section describes the experiment we carried on as a preliminary exploration of the subject and verification of feasibility of our proposal. We developed a taxonomy of interest entities in the domain called Legal, and developed a prototype intended for Named Entity Recognition in the chosen domain. Furthermore, we executed this prototype over a corpus with privacy law and guidelines texts, and the results are detailed in this section also.

3.1 Legal ontology

Several taxonomies have been proposed in order to classify entities in the Named Entity Recognition task (Brunstein, 2002; Sekine, 2008; Linguistic Data Consortium, 2008). Yet, they did not show specificity enough to represent the subject we research fully, in such a way we had to specialize and extend classes of interest, and prone others that showed to be less significant for us right now.

The Legal ontology was built as a way to classify entities of interest referring to the legal domain, plus privacy and accountability actions, goals and risks. It models norms and regulations specific to data privacy.

It was constructed manually, supported by the domain study and the comparison with other taxonomies for NER and other ontologies referring to law and privacy (Hoekstra *et al.*, 2007; Silva *et al.*, 2009).

It is also essentially different from other legal ontologies in the sense that it is not intended to follow a functional or rigid ontological approach (such as FOLaw or LKIF) (Breuker & Hoekstra, 2004; Hoekstra *et al.*, 2007). It is intended only to classify interest named entities, allowing them to be linked in relations later in a flexible way. The Legal ontology does not represent the legal domain entirely, and neither has it as a goal. However, as its instances and relations fill it, we can perceive its power as the basis for a tool aimed at project managers in the software industry. Figure 1 presents the current taxonomy of Legal, which encompasses 21 classes total. The top four classes are: Regulation, Resource, Theme and Geo, being the first two the most significant ones.

Instances of Regulation are any abstract entity which dictates rules for individuals to follow under certain circumstances, named previously in other NER taxonomies as OBRA/PLANO [1] (Santos & Cardoso, 2007), NAME/PRODUCT/RULE (Sekine, 2008) and Norm (Hoekstra *et al.*, 2007). The class Resource is intended for resources which document regulations, such as an URL. Theme is the theme an instance of Regulation talks about (examples are "transborder data flow" and "health information").

As for the Geo class, we do not intend at the present moment to explore it more deeply ourselves. Geographic ontologies are the main subject of very exciting previous work (Vatant & Wick, 2006) done by other research groups, and it is our intention to reuse part of this work in our Legal ontology instead of re-inventing it.

3.2 Prototype

The system presented in this section was designed as part of a feasibility study of the NER task aiming the population of an ontology. Its architecture is illustrated in Figure 2.

The NER module identifies Law, Act and Rule entities and classifies them. After that, passes the entities' list for OntoPopulate, which populates the received taxonomy with the entities as instances.

Python[2] was used for prototyping the system, which uses also NLTK[3] for sentence splitting, tokenization and

[1] Portuguese for "WORK/PLAN". This classification is intended for use in a Portuguese NER evaluation, and therefore uses Portuguese classes' names.

[2] http://python.org

[3] Natural Language Toolkit, available at http://www.nltk.org.

POS tagging. The NER module of NLTK was not used, but instead a method developed by us for this task.

Our method for NER is currently restricted to entities of Law, Rule and Act (as a preliminary experiment). The method involves the identification of syntactic and positional patterns, and we intend as a next step to extend it with the addition of semantics using resources as Wordnet and domain ontologies.

In the first place, the system looks for specific keywords in the corpus, searching for laws. These keywords are taken from Legal ontology: those which are specifically related to laws and regulations are chosen, seeding the corpus search process. Currently, three keywords are first searched: *act*, *rule* and *law*. These are the kernel of the searched patterns.

If the markup given by NLTK for the found word is not a verb (intended to exclude conjugations of "to act" or "to rule", for instance), the next verifications take place. These verifications include a search for determiners (the, this) and identifiers (numbers, year and capitalized qualifiers).

The entity is always limited by the end of the sentences, or before. No entities include more than one sentence. NLTK is used in order to separate the sentences. It delimits the sentences based mostly in period punctuation.

Other cases, when the end of the recognized entity is delimited before the end of the sentence, are the two following:

1. A number (that may be a year or identifier, according to our corpus study) follows the keyword (as in "Law 15/1999");
2. A number preceded by "of" follows the keyword (as in "Act of 2003").

The delimitation of the beginning of the entities obeys one of the following cuts:

1. A determiner ("the" or "this", in the retrieved results), which is processed and used in order to delimit the entity but not included in it. An example of sentence from which the entity was delimited this way is "This Act may be cited as the **Spam Act 2003**", in the Australian Spam Act of 2003.
2. A coordinating conjunction such as "or" and "and", intended for separating different laws in a listing of them. No cases for this pattern were found in our corpus.
3. Size of the entity. An arbitrary number of 10 tokens for each entity was set up and showed to be effective. This way, some bigger and incorrect entities resulted from the corpus formatting (tables and indexes, for instance) were removed. One example is "...Giving effect to international conventions 35 46 Review of operation of **Act 35**".

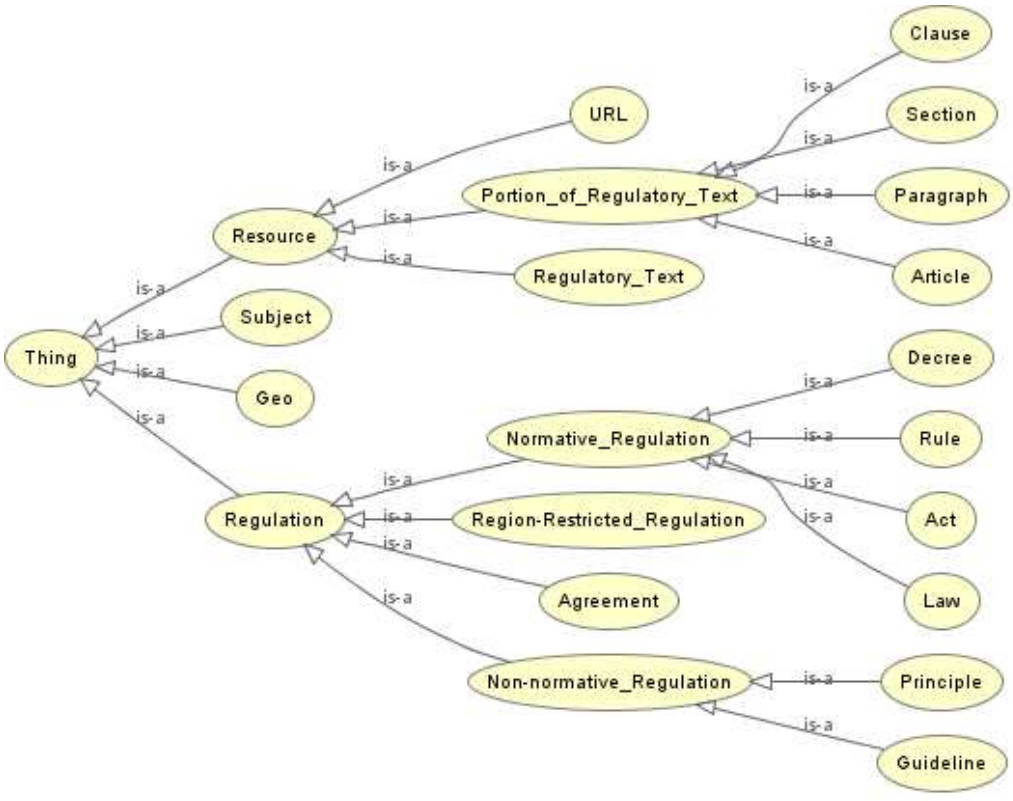

Figure 1. Taxonomy of Legal, designed to classify entities of the legal domain, especially on privacy and accountability matters.

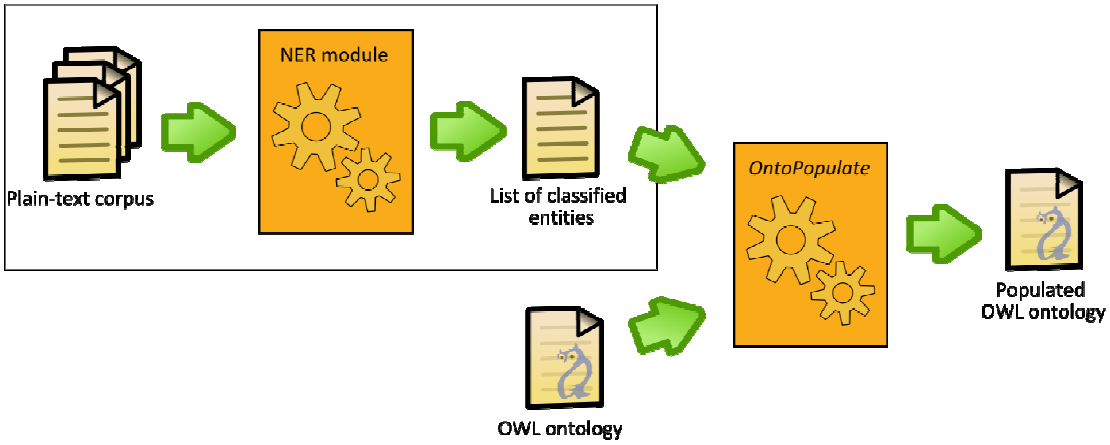

Figure 2. Architecture of the prototype which populates an ontology from text.

After the identification, minor corrections are performed over the entities, such as the removal of surrounding instances of pipeline character ("|"; used for formatting of tables in some source texts) and extra spaces between parenthesis.

Duplicated instances of the entities are unified and populate the ontology only once.

Example entities that are identified and populate the ontology are: "People's Republic of China Telecommunication Rule", "Organic Law 15/1999" and "Spam Act of 2003".

3.3 Evaluation and results

The evaluation was carried on a corpus of 25 texts and approximately 200,000 words composed by different types of documents, from legislation on privacy in various countries to software industry guidelines. These texts were selected due to their relevance on the privacy legislation area in several countries. Legislation from European Union and the United States is the majority of our corpus.

Comparable results are obtained from the 2008 ACE Local Entity Detection and Recognition (EDR) Evaluation. However, this evaluation (or any other that we know of) does not provide a corpus in the domain we intend to work on, which is the main reason we built our own corpus for that.

The execution of the prototype over this corpus resulted in 128 references to named entities of interest, and 59 unique entities. Precision and recall were computed in comparison to a manual annotation of the same corpus and are presented in Table 1. The numbers which are comparable to those of the ACE Local EDR track are the ones listed in "References to entities", since ACE tracks consider all mentions to entities in their results.

The best results presented in ACE Local EDR track are 52.6% (Linguistic Data Consortium, 2008b), considering only the common classes for entity detection and classification: Person, Organization, Location and so on.

	Precision	Recall	F-Measure[4]
References to entities[5]	79.69% (102/128)	21.21% (102/481)	33.49
Unique named entities	66.10% (39/59)	31.71% (39/123)	42.86

Table 1: Numerical results for the execution of the developed prototype for ontology population from legal texts using Named Entity Recognition.

These are promising results, considering the diverse and innovative nature of the studied entity classes (Laws and others) and how much more could be achieved in larger corpora and with more sophisticated techniques aiding the NER in the task of ontology population. These techniques could improve significantly our results, specially the recall measure, which is still quite low. We believe this happens mainly due to an insufficient number of heuristics to detect entities. Section 5 enumerates these conclusions and future directions on this work in order to solve these issues.

4. Related work

Much work has been devoted to ontology learning tasks since the beginning of the decade, when attention has turned to ontologies and semantic applications.

One of the first important works in the area is that presented by Maedche & Staab (2001), called Text-To-Onto. Their work is justified by them arguing that the reduction of the difficulty of the knowledge acquisition task is a requirement for the success of the Semantic Web.

Text-To-Onto is an environment that provides support for the ontology engineer in all the stages of ontology construction: import, extraction, pruning, refinement and even evaluation. Among other features, the system supports ontology learning from free text, which is a similar approach to that presented in our proposal. However, Text-To-Onto is intended to assist the ontology engineer in all the aspects of the creation of the

[4] F-Measure in this paper is calculated by `(2*Precision*Recall)/(Recall+Precision)`

[5] Including duplicates.

ontology, and our intention is populate an already existing ontology.

There have been also work proposals focused in the legal domain. Lame & Desprès (2005) present a whole set of techniques that may be used in order to automatically update a previously built ontology. Their motivation is related to the laws' constant changes and the need for ontologies that can follow these changes. This is the scenario they present in their experiment: two ontologies, built in different moments and therefore result of different versions of the law, are intended to be merged and result in an updated ontology. Lame & Desprès align concepts and relations of the two ontologies and the result is the final one. They focus on NLP-based techniques, both syntactical and statistical.

Lenci *et al.* (2009) report an experiment on their ontology learning system called T2K. They mix the use of NLP techniques with Machine Learning in order to extract terms and relations from free text. The experiment conducted by them uses Italian legal texts and correctly identify classes for the ontology, as well as many hyponymy relations (illustrated in the paper).

The research presented by Peters (2009) is proposed in order to enrich already built ontologies, and uses legal texts for that. The author presents an exploratory study on automatic and semiautomatic NLP-based techniques aiming ontology enrichment.

The process of ontology enrichment, according to the author, involves two aspects: new terms and new relations, emerging from the data available and used for the updating process. The experiment is conducted using the GATE platform and presents good results (81.2% average).

5. Final remarks and future work

In this paper, we presented an experiment for ontology population from legal and normative texts through the task of Named Entity Recognition.

Even preliminary, the quantitative and qualitative results shown so far are promising (relatively low recall, but high precision and very accurate resulting instances in the populated ontology).

We attribute the low recall to the small number of heuristics used to detect the entities. Currently, new approaches are being experimented and added to our system, aiming the increase of the amount of retrieved entities. Besides that, we also intend to expand the corpus, adding new law and doctrinal texts in the subject of privacy.

However, even in its current state, the experiment on this system showed the feasibility of NER aiding ontology population, and we believe that additional resources (such as Wordnet and domain ontologies) and techniques (such as more syntactic and positional patterns) in use of our system will present even better results. This is currently being done in this research. Also, we intend to experiment on the extraction of semantic relations between the entities recognized in this experiment, starting from those which relate different instances of laws (like when some document refers to another) and those which relate region-specific laws to its geopolitical entities, reusing a geographical ontology.

Our research in the area is motivated as an aiding mechanism for compliance checking of actions in projects with current laws and regulations. The hard task of identifying law breaches can be aided by automatic NLP, and such applications can be used both in organizations and government for their products and services.

6. Acknowledgements

This paper was achieved in cooperation with Hewlett-Packard Brasil Ltda. using incentives of Brazilian Informatics Law (Law nº 8.2.48 of 1991).

7. References

Ajani, G., Boella, G., Lesmo, L., Martin, M., Mazzei, A., Radicioni, D.P., Rossi, P. (2009). Legal Taxonomy Syllabus version 2.0. In *Proceedings of the Third Workshop on Legal Ontologies and Artificial Intelligence Techniques*, v. 465. Barcelona, Spain: CEUR-WS.

Breuker, J., Hoekstra, R. (2004). Epistemology and ontology in core ontologies: Folaw and LRI-Core, two core ontologies for law. In *Proceedings of the EKAW'04 Workshop on Core Ontologies in Ontology Engineering*. Northamptonshire, UK.

Brunstein, A. (2002). Annotation guidelines for answer types. Available at: http://www.ldc.upenn.edu/Catalog/docs/LDC2005T33/BBN-Types-Subtypes.html.

Cybenko, G. (2006). Why Johnny Can't Evaluate Security Risk. *IEEE Security and Privacy* 4, 1 (Jan. 2006), 5.

Denker, G., Kagal, L., Finin, T., Sycara, K., Paoucci, M.: (2003). Security for DAML web services: Annotation and matchmaking. Berlin / Heidelberg: Springer.

Vatant, B., Wick, M. (2006). Geonames ontology. http://www.geonames.org/ontology/.

Hoekstra, R., Breuker, J., Di Bello, M., Boer, A., (2007). The LKIF Core ontology of basic legal concepts. In Casanovas, P., Biasiotti, M.A., Francesconi, E., and Sagri, M.T. (eds.), *Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques* (LOAIT 2007).

Kagal, L., (2002). A policy language for the Me-centric project. Technical Report HPL-2002-270, HP Laboratories.

Kagal, L., Weitzner, D. J., Hanson, C., (2008). Using dependency tracking to provide explanations for policy management. In *POLICY'08*.

Lame, G., Desprès, S. (2005). Updating ontologies in the legal domain. In *Proceedings of the 10th international Conference on Artificial intelligence and Law* (ICAIL'05). New York, NY: ACM, pages 155-162.

Lenci, A., Montemagni, S., Pirrelli, V., Venturi, G. (2009). Ontology learning from Italian legal texts. In *Proceeding of the 2009 Conference on Law, ontologies and the Semantic Web: Channelling the Legal*

information Flood. In J. Breuker, P. Casanovas, M. C. Klein, and E. Francesconi, (Eds). Frontiers in Artificial Intelligence and Applications, v. 188. Amsterdam, The Netherlands: IOS Press, pages 75-94.

Linguistic Data Consortium (2008). ACE english annotation guidelines for entities.

Linguistic Data Consortium (2008b). NIST 2008 Automatic Content Extraction Evaluation (ACE08) Official Results. Available at: http://www.itl.nist.gov/iad/mig/tests/ace/2008/doc/ace08_eval_official_results_20080929.html.

Maedche, A. Staab, S. (2001). Ontology Learning for the Semantic Web. *IEEE Intelligent Systems* 16 (2), pages 72-79.

Mont, M., Thyne, R. (2006). Privacy policy enforcement in enterprises with identity management solutions. In: *PST '06*, vol. 380, pages 1–12. New York: ACM.

Peters, W., (2009). Text-based Legal Ontology Enrichment. In *Proceedings of the Third Workshop on Legal Ontologies and Artificial Intelligence Techniques*, volume 465. Barcelona, Spain: CEUR-WS.

Santos, D., Cardoso, N. (2007). Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área. Linguateca. ISBN: 978-989-20-0731-1.

Sekine, S. (2008). Extended named entity ontology with attribute information. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., e Tapias, D. (Eds.), *Proceedings of the Sixth International Language Resources and Evaluation* (LREC'08), Marrakech, Marrocos.

Silva, D., Bruckschen, M., Bridi, P., Granada, R., Vieira, R., Agustini, A., Northfleet, C., Rao, P., Sander, T. (2009). Semantic Web and Knowledge Management in User Data Privacy. In *ER'09: 28th International Conference on Conceptual Modeling (Poster session)* (to appear), Gramado, Brazil.

Legal Claim Identification: Information Extraction with Hierarchically Labeled Data

Mihai Surdeanu, Ramesh Nallapati and Christopher Manning

Stanford University
{mihais,nmramesh,manning}@cs.stanford.edu

Abstract

This paper introduces a novel Information Extraction problem, where only parts of documents have relevance and linguistic annotations are available only for these segments. The data is hierarchical: the top layer marks the relevant text segments and the bottom layer annotates domain-specific entity mentions, but only in the segments marked as relevant in the top layer. We investigate this problem in the legal domain, where we extract the text corresponding to litigation claims and entity mentions such as patents and laws in each claim. Because entity mentions are not labeled outside claims in training data, a top-down approach that extracts claims first and entity mentions next seems the most natural. However, we show that other models are superior. Using a simple semi-supervised approach we implement a bottom-up Conditional Random Field model; we also implement a joint hierarchical CRF using a combination of pseudo-likelihood and Gibbs sampling. We show that both these models significantly outperform the top-down approach.

1. Introduction

Most state-of-the-art supervised Information Extraction (IE) approaches can be classified in two classes: *flat* extractors, which segment text into relevant regions, e.g., named entity mentions (Sang and Meulder, 2003) or elements of seminar announcements (Freitag, 1998), or *deep* extractors, which construct complex domain-specific semantic representations of content, e.g., the scenarios proposed by the Message Understanding Conference (MUC)[1] or the events and relations promoted by the Automatic Content Extraction (ACE) evaluations[2]. While the latter class of approaches are closer to true natural language understanding, such systems have not yet achieved commercial acceptance due to their relatively poor performance.

In this paper we argue that representations of intermediate complexity are more attractive for practical applications. Motivated by a real-world IE domain, we propose a novel IE task composed of two subtasks or layers: in the first layer we extract text segments relevant to the given domain and in the second layer we extract important entities[3] from these segments. Figure 1 shows a hypothetical example with such annotations. An important observation is that, for practicality, we implement a hierarchical annotation process, i.e., entities are annotated only inside regions of interest. This essentially yields an asymmetric task: while the top layer is fully annotated, the bottom layer has only partial annotations, i.e., many entities outside relevant regions are left unlabeled.

There are many domains where such a framework is useful. For example, somebody interested in the 2008 Olympic Games may want to extract only the relevant passages and corresponding entities from articles about Beijing, e.g., players, venues, dates, etc. Technology-savvy blog readers may be interested only in blog passages related to technology and entities such as gadget names and prices. In this paper, we focus on a third domain: Intellectual Property (IP) litigation, where we extract the text corresponding to litigation claims from pleading documents and the relevant entities inside each claim, e.g., patents and laws (see Figure 2 for an example). This task is motivated by several immediate applications: case summarization, semi-structured search inside claim texts, structured search over claim entities, visualization of the inter-party relations, e.g., who infringes whose patent.

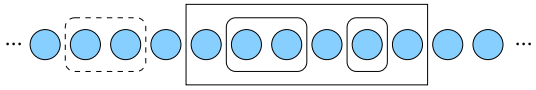

Figure 1: An example of text with hierarchical annotations. Individual words are circles, relevant text regions are rectangles, and the embedded entity mentions are rectangles with rounded corners. Entity mentions also occur outside of regions of interest and are represented here with dashed lines, i.e., they are unlabeled.

The contribution of this paper are two fold:

- We introduce a novel IE task motivated by a real-world application. We evaluate the constructed systems on a legal domain using data from actual case documents. The data is noisy: it comes from PDF documents converted automatically to text or from scanned documents converted to text using an Optical Character Recognition (OCR) system.
- Although the hierarchical nature of the task seems to impose a top-down approach, we show that other less intuitive models are preferable. Using a simple semi-supervised approach that addresses the missing labels in the entity layer we implement a bottom-up Conditional Random Field (CRF) (Lafferty et al., 2001) model. We also implement a joint hierarchical CRF model that extracts the two layers jointly using a combination of pseudo-likelihood and Gibbs sampling. We show that both these models outperform the top-down approach significantly.

The paper is organized as follows. Section 2 describes the IE task with a focus on the legal domain. Section 3 introduces the proposed models. Section 4 shows the results

[1] http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_toc.html

[2] http://www.itl.nist.gov/iad/mig//tests/ace/

[3] Throughout the paper we will use "entities" to stand for "entity mentions", for brevity.

...
31. On February 20, 2007, the USPTO duly and legally issued United States Pa\tent No. 7,179,046 B2 ("the '046 patent"), also entitled "Fan array fan section in air-handling
8

systems." Huntair is the owner by assignment of all right, title and interest in and to the '046 patent. A copy of the '046 patent is attached to the Complaint as Exhibit A.
{ClaimBegin [**FIRST COUNTERCLAIM**]ClaimNumber [**INFRINGEMENT**]ClaimType OF [**U.S. PATENT NO. 7,137,775 B2**]Patent 32. Huntair repeats and realleges paragraphs 26-31 as though fully set forth\ herein.
33. Upon information and belief, Plaintiff [**is and continues to be directly infringing, contributorily infringing, and/or inducing infringement**]ClaimType of the [**'775 patent**]Patent by, among other things, making, using, offering to sell, selling and/or
importing, without authority or license from
Plaintiff, fan arrays in this district and elsewhere in the United States, which embody, incorporate, or otherwise practice one or more claims of the [**'775 patent**]Patent.
34. Upon information and belief, in its bid to obtain a contract to install an array of\
fans at facilities owned by Amcol in Chicago, Illi
nois, Plaintiff offered to utilize a fan system
that contains, embodies, and employs the invention described and claimed in the [**'775 patent**]Patent.
35. Plaintiff's conduct constitutes infringement, as provided by [**35 U.S.C. $ 271**]Law, of
one or more claims of the [**'775 patent**]Patent.
36. As a result of this infringement, Huntair has been damaged and deprived of the
gains and profits to which it is entitled. Furthermore, Huntair will continue to be damaged unless
this Court enjoins Plaintiff's infringing conduct.ClaimEnd}
{ClaimBegin [**SECOND COUNTERCLAIM**]ClaimNumber [**INFRINGEMENT**]ClaimType [**OF U.S. PATENT NO. 7,179,046 B2**]Patent 37. Huntair repeats and realleges paragraphs 26-31 as though fully set forth herein.
...

Figure 2: A representative example of an annotated pleading document from an IP litigation case. Claim boundaries are marked with {ClaimBegin and ClaimEnd}. Claim entities are in bold face and delimited by squared parentheses, e.g., [...]Patent. Party names are not annotated because they are available in the case meta data.

of our empirical evaluation. Section 5 summarizes related work and Section 6 concludes the paper.

2. Problem Description

We start this section with a description of the IP litigation domain, as a concrete instance of the proposed IE task. Figure 2 shows an example annotated document from this domain. Other than adding the annotation labels and using bold face for entity mentions we preserved the format of the original document. The figure illustrates several of the issues that plague this data: incorrect pagination, e.g., new paragraphs created in the middle of sentences, missing or extraneous characters, e.g., *"Pa\tent"*, broken words, e.g., *"Illi nois"*, etc.

The domain has two layers of annotations. In the top layer we annotate the claim text regions, shown between {ClaimBegin and ClaimEnd} in the figure. The claim segments contain all the text that is vital to understand the claim (e.g., who infringes which patent) but no extraneous material (e.g., background information about the parties involved in the case or the relief sought). Ideally, these are separated sections in a pleading document, but in practice, it is common that this information be mixed. This makes the processing of pleading documents a non-trivial process, and is further motivation for an automated extraction system. The bottom layer annotates important entities inside claims:

Patent (P) – contains references to patent numbers, such as *"United States Patent No. 6,190,044"* or *" '044 patent"*.

Law (L) – marks references to both federal and state laws, including sections and sub-sections, e.g., *"35 U.S.C. $ 281, 283, 284, and 285"* or *"California 7 Business & Professions Code $ 17200, et seq."*. Here the *$* sign is a typical error of our pre-processing system, which often fails to recognize the section mark symbol (§).

ClaimNumber (N) – annotates the numbered header that usually marks the beginning of the claim, e.g., *"First cause of action"*, *"Second claim for relief"*. These headers uniquely identify a claim, but they are often missing.

ClaimType (T) – identifies the type of the parent claim. It is typically instantiated by verbal phrases or verb nominalizations (see figure). These are obviously not entity mentions; they are more reminiscent of ACE event anchors. However, for brevity, we will refer to all these four segment types as "entities" throughout the paper.

From this domain definition we drew several important observations that drove the design of our IE models. First, because the relevant text segments (e.g., claims) are likely to cover several sentences or paragraphs, the extractors in the top layer must model the text at a granularity larger than individual words. As a proof of concept we ran a state-of-the-art Conditional Random Field (CRF) sequential tagger trained at word level for the task of extracting the claim regions. The performance was very low: approximately 5 F_1 points.[4] Based on this observation, we design our extractors for the top layer to use sentences as the atomic elements. Second, although entities can occur both inside and

[4] We detail our evaluation metrics in Section 4.

outside relevant text regions, during training entity tags are only available for sentences that are tagged as belonging to a segment of interest (e.g., claim). This was done because typically the entities of interest in the given domain are the ones mentioned inside relevant text regions (e.g., we are only interested in the infringed patents) and focusing on this content saves significant annotation effort[5]. This indicates that the most natural approach for this task follows a top-down architecture: first extract claim segments, and then extract the relevant entities from these claims. And finally, entities occur outside relevant text regions as well and it is reasonable to assume that they occur in stylistically similar text (after all, it is written by the same person) and some of the context is shared (Figure 2 shows that some of the claim patents are mentioned outside as well). Hence there is potential benefit in modeling the entities outside claims as well. This motivates our semi-supervised model introduced in the next section.

3. Models

In the following subsections, we will describe several architectures that model this problem, starting with the simplest first. All the architectures use Conditional Random Fields (Lafferty et al., 2001) as a fundamental building block. We model both layers using first-order CRF taggers, using the Begin (B) – Inside (I) – Outside (O) notation to mark relevant segments in both layers, i.e., 'B' is assigned to elements (sentences or words, depending on the layer) that begin a relevant segment, 'I' is assigned to other elements inside the segment, and 'O' labels elements outside any relevant snippet.

In the top layer, the claim tag for each sentence s is represented by a discrete random variable C_s, and it takes values from the set $\{B, I, O\}$. We also denote the sequence of claim tags in a given document d by the vector $\mathbf{C}_d$. In the entity layer, $E_i \in \{\{B, I\} \times \{N, T, P, L\}\} \cup \{O\}$ represents the entity tag for the word at i^{th} position in a sentence. In other words, each word can be in the beginning ('B') or inside('I') of one of the four entity types or just be a non-entity (captured by the 'O' tag). We also represent the sequence of entity tags in a given sentence s by $\mathbf{E}_s$. $\mathbf{X}_d$ denotes the entire document text while $\mathbf{X}_s$ represents the text in sentence s, and X_i represents the i^{th} word in that sentence. We will use lower case letters to denote the values assumed by random variables (*e.g.*: c, $\mathbf{e}$, and x for a claim, an entity sequence and a textual token respectively). In addition, we use bold faced notation to represent sequences and regular faces to represent singleton tokens (*e.g.*: $\mathbf{C}$ for claim tag sequence and C for a singleton claim tag). We will omit subscripts where it is clear from the context.

3.1. Top-Down CRF

The top-down CRF is a simple architecture that closely mirrors the annotation process. In this approach, we train two independent CRFs which we call Claim CRF and Entity CRF. The Claim CRF operates on the whole document and considers each sentence as the smallest unit. It models the probability of claim tags sequence $\mathbf{C}_d$ for the document d conditioned only on text $\mathbf{X}_d = \mathbf{x}_d$, represented as $P(\mathbf{C}_d|\mathbf{x}_d)$.

The Entity CRF operates at the sentence level and considers each word as its smallest unit. For each sentence s, the Entity CRF models $P(\mathbf{E}_s|\mathbf{x}_s, c_s)$, the probability of its entity tag sequence $\mathbf{E}_s$ conditioned on the sentence text $\mathbf{x}_s$ as well as the corresponding claim tag $C_s = c_s$. The Entity CRF trains only from data inside claims because there is no labeled data available for entities outside claims.

At inference time, we first run the Viterbi algorithm for inference on the Claim CRF to generate the predicted claim sequence $\mathbf{c}_d^{(p)}$ for the whole document d. Then, we run inference for Entity CRF on each sentence s labeled as 'B' or 'I' by the Claim CRF, conditioned on the text $\mathbf{x}_d$, to output its predicted entity tag sequence $\mathbf{e}_s^{(p)}$.

The top-down model can be visualized from Figure 3, which displays a generic representation of all models discussed in this paper. The broken arrows from claims to entities in the figure correspond to this model and represent flow of information from claims to entities.

The probabilities modeled by the Claim CRF and the Entity CRF, and the inference order are summarized in row 1 of Table 1.

3.2. Bottom-up CRF

In the previous approach, the Claim CRF is ignorant of the underlying entities in the next layer. It is conceivable that the performance of the top layer Claim CRF could be improved by transmitting to it the entity information in each sentence, e.g., it is more probable to see references to patent numbers or statutes inside claim texts.

As a natural first approach, we use a bottom-up architecture as follows: for each sentence s, the Entity CRF models the probability of the entity sequence $\mathbf{E}_s$ conditioned only on the observed text sequence $\mathbf{x}_s$, given by $P(\mathbf{E}_s|\mathbf{x}_s)$. The Claim CRF, on the other hand, models for each document d, $P(\mathbf{C}_d|\mathbf{x}_d, \mathbf{e}_d)$, the probability of the claim sequence $\mathbf{C}_d$ conditioned on the entire observed document text $\mathbf{x}_d$ and the entity tag sequence of the entire document $\mathbf{e}_d$.

At inference time, we first run inference on the entity sequence using the Entity CRF to produce predicted entity tag sequence $\mathbf{e}_d^{(p)}$ and then run inference on the Claim CRF conditioned on these entity tags, to generate the predicted claim tag sequence $\mathbf{c}_d^{(p)}$. As a post-processing step, we remove the entity tags $\mathbf{e}^{(p)}$ that are outside the claims to output the final entity tags $\mathbf{e}_d^{(\text{constraints})}$. This additional cleaning up process for entities is necessitated in the bottom-up approach to satisfy the problem constraints that entities occur only inside claims.[6] The exact models for claims and entities for this architecture, and the inference order are displayed in row 2 of Table1.

This model will result in inferior performance owing to the missing entity labels outside claims. To elaborate, since the Entity CRF in this bottom-up architecture is oblivious

[5] A latent assumption is that most of the text is outside claims. This is why there are significant savings in not marking entities outside claims.

[6] Recall that in the top-down approach, the Entity CRF was conditioned on the claim tags, so it would learn to label entities only inside claims.

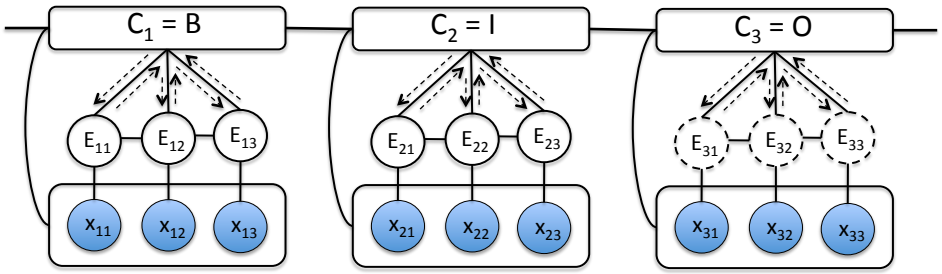

Figure 3: Generic graphical representation of all the models discussed in this paper: the top nodes represent the claim layer, the middle layer represents the entity layer and the bottom layer is text. Each node in the entity layer corresponds to a word, while each node in the claim layer corresponds to a sentence. The text nodes are darkly shaded because they are observed. The broken entity nodes in the third sentence, labeled in the top layer as outside claim ('O'), indicate that, outside claim mentions, entities are unlabeled at training time and ignored at test time. The edges correspond to the dependencies captured by the model (we removed some non-essential edges to prevent clutter). There are three types of edges between claims and entities: (a) the broken arrows from claims to entities represent the top-down pipelined system, (b) the broken arrows from entities to claims represent the two bottom-up pipelined systems and (c) the solid undirected edges represent the joint hierarchical model.

	Architecture	**Claim Model**	**Entity Model(s)**	**Order of Inference**
1	Top-down	$P(\mathbf{C}_d\|\mathbf{x}_d)$	$P(\mathbf{E}_s\|\mathbf{x}_s, c_s)$	$\mathbf{c}^{(p)} \rightarrow \mathbf{e}^{(p)}$
2	Bottom-up	$P(\mathbf{C}_d\|\mathbf{e}_d, \mathbf{x}_d)$	$P(\mathbf{E}_s\|\mathbf{x}_s)$	$\mathbf{e}^{(p)} \rightarrow \mathbf{c}^{(p)} \rightarrow \mathbf{e}^{(\text{constraints})}$
3	Semi-sup. Bottom-up	$P(\mathbf{C}_d\|\mathbf{e}_d^{(\text{semi})}, \mathbf{x}_d)$	$P(\mathbf{E}_s\|\mathbf{x}_s)$	$\mathbf{e}^{(p)} \rightarrow \mathbf{c}^{(p)} \rightarrow \mathbf{e}^{(\text{constraints})}$
4	Semi-sup. Joint Hierarchical	$P(\mathbf{C}_d\|\mathbf{e}_d^{(\text{semi})}, \mathbf{x}_d)$	$P(\mathbf{E}_s\|\mathbf{x}_s)$, $P(\mathbf{E}_s^{(\text{semi})}\|\mathbf{x}_s, c_s)$	$\mathbf{e}^{(p)} \leftrightarrow \mathbf{c}^{(p)} \rightarrow \mathbf{e}^{(\text{constraints})}$

Table 1: Various architectures and their corresponding models.

to the claim information, at inference time, it is free to assign entity tags $\mathbf{e}^{(p)}$ in any sentence irrespective of its claim tag. Furthermore, since the Claim CRF is conditioned on labeled entities at training time, and since there are no labeled entities outside claims in training data, it learns that sentences that contain entities are very likely to be claims. Hence, performing inference on the Claim CRF conditioning on $\mathbf{e}^{(p)}$ may result in a large number of false positives for claims. In the next subsection, we will present a modified bottom-up architecture that will address the problem of missing labeled data in the entity layer.

3.3. Semi-supervised Bottom-up CRF

The bottom-up approach is problematic because the hierarchical nature of labeling generates partial entity labels in the annotated data, which may inject an unreasonable bias in the Claim CRF. If entity labels were available outside claims, the Claim CRF conditioned on entities would learn the true correlation between the presence of entities and claim segments. Hence, in this approach, we first train the Entity CRF only on sentences labeled as claims, and run it on the entire training set to generate predicted labels $\mathbf{e}^{(p)}$. We augment the labeled entities $\mathbf{e}$ from inside claims with $\mathbf{e}^{(p)}$ outside the claims to generate our semi-supervised labeled entity sequence $\mathbf{e}_d^{(\text{semi})}$. We use this data to condition the Claim CRF at training time.

Thus, the only difference between the semi-supervised bottom-up approach and the bottom-up approach is that the Claim CRF trains on semi-supervised entity labels $\mathbf{e}^{(\text{semi})}$ instead of only gold entity labels $\mathbf{e}$ as shown in row 3 of Table 1. Both these models are represented in Figure 3 by the broken arrows pointing upwards, symbolizing the pipelined information flow from entities to claims.

Since this model uses entities both inside and outside claims, it can be expected to capture the true correlation between entities and claims better than the standard bottom-up approach. An additional boost in performance may be expected also because the Claim CRF, training on predicted entities, can learn additional contextual and stylistic features of entities from outside the claims. Note that the standard bottom-up CRF presented above did not have this advantage.

3.4. Semi-supervised Joint Hierarchical CRF

The pipelined approaches discussed thus far model only one-way flow of information from one layer to the other. It is reasonable to assume that there is potential benefit in modeling both the layers jointly: the Entity CRF could recognize the relevant entities better, knowing whether it is inside or outside a claim, while the Claim CRF could tag the claims better, knowing what type of entities are more likely to occur inside claims than outside.

The new model therefore estimates the joint probability of both $\mathbf{C}_d$ and $\mathbf{E}_d$, conditioned on the observed document text sequence $\mathbf{x}_d$. The graphical representation of this new model is shown in Figure 3 as solid undirected edges between claims and entities. The model is hierarchical by definition because the top layer of claims is at sentence level while the bottom layer is at word token level.

Although this model is more attractive than the pipelined models, exact learning is practically infeasible [7]. Hence, in

[7] The complexity of inference is $O((|L_1| \times |L_2|)^2 n)$, where L_1 is the label set for the top layer and L_2 is the label set for the bottom layer and n is the length of the sequence.

this paper, we use a variant of pseudo-likelihood for training (Besag, 1975). Pseudo-likelihood is known to be a consistent estimator of true likelihood and is known to work well in cases where local features are strong (Parise and Welling, 2005; Toutanova et al., 2003). In this method, the joint likelihood of all the variables in a model is approximated by the product of the probability of each variable, conditioned on all other variables. In our model we apply the pseudo-likelihood only between the two layers as shown below:

$$P(\mathbf{C}, \mathbf{E}|\mathbf{x}) \approx P(\mathbf{C}|\mathbf{E}, \mathbf{x})P(\mathbf{E}|\mathbf{C}, \mathbf{x}) \quad (1)$$

This approximation makes learning efficient because each conditional probability in the right hand side of Eqn. 1 reduces to two conditional CRFs: $P(\mathbf{C}|\mathbf{E}, \mathbf{x})$ is the Claim CRF conditioned on entities while $P(\mathbf{E}|\mathbf{C}, \mathbf{x})$ is the Entity CRF conditioned on claims, both of which can be estimated using exact methods for CRFs.

Similar to the semi-supervised bottom-up approach, we train the Claim CRF $P(\mathbf{C}|\mathbf{E}, \mathbf{x})$ conditioned on semi-supervised entity labels $\mathbf{e}^{(\text{semi})}$ as shown in row 4 of Table 1. The symmetric nature of the joint model leaves us no choice but to train the Entity CRF also on $\mathbf{e}^{(\text{semi})}$, as shown in the same row of Table 1. We also list an unconditioned Entity CRF $P(\mathbf{E}|\mathbf{x})$ as an additional model used in this architecture because it is required to generate $\mathbf{e}^{(\text{semi})}$ at training and $\mathbf{e}^{(p)}$ at testing time.

Since exact inference is computationally expensive as well, we use Gibbs sampling (Andrieu et al., 2003) to perform approximate inference, since it has many interesting parallels with pseudo-likelihood. Like pseudo-likelihood, Gibbs sampling deals with local probability of each variable, conditioned on all other variables.[8] In this approach, we sample each variable in turn from its probability conditioned on its latest assignments of its neighbors. This iterative process, when run long enough is guaranteed to converge to the true posterior.

In our case, since we have a two tier hierarchy, in each iteration, we successively sample all the variables in one layer then move to the other layer. Also, since we need best variable assignments rather than true posterior, we use *simulated annealing* with Gibbs sampling, using a linear *cooling schedule*, as proposed in (Finkel et al., 2005).

4. Experimental Results

We start this section by describing the experimental settings, we continue with a description of the feature set used in both subtasks, and we conclude with a discussion of the experimental results.

4.1. Data

The corpus used in this paper contains 90 pleading documents from actual IP litigation cases. The documents are either PDF documents converted to text (for newer cases) or scanned documents converted to text using an OCR system (for older cases). A significant amount of noise was introduced in the data by this process. The corpus was preprocessed using an in-house tokenizer and sentence boundary detector. The sentence boundary was adapted to the pagination of this corpus, e.g., it introduces sentence breaks at two consecutive new line characters even if no punctuation mark exists. The resulting tokenized text was part-of-speech (POS) tagged using the Stanford POS tagger[9]. Lastly, the corpus was annotated by an IP litigation expert, who followed strict annotation guidelines designed by a multi-disciplinary group of experts from both Law and Computer Science. Table 2 summarizes the corpus statistics.

This corpus was randomly split into a training partition (70%) and a testing partition (30%). We were careful not to have documents from the same case in both training and testing.[10] This yielded a training corpus of 64 documents and a testing set of 26 documents.

4.2. Evaluation Metrics

As evaluation metrics we used the standard precision, recall, and F_1 scores coupled with a strict-match criterion in the spirit of the CoNLL evaluations (Sang and Meulder, 2003). In other words, an extracted segment is considered correct if it matches exactly the tokens in the corresponding annotation and it has the correct label.

4.3. Features

For Entity CRF we used a modified version of the Stanford Named Entity Recognition (NER) software[11] (Finkel et al., 2005). We used its default feature set consisting of: (a) word, (b) part of speech (POS) tag, and (c) word-shape, where the word shape captures the case of the alpha characters in the word, collapses sequences of the same type, but maintains punctuation. These features are extracted from the current word and its immediate context, i.e., the previous and following word. We extended this feature set with only one new feature: the claim tag of the current sentence c_s (for the top-down and joint approaches).

For Claim CRF, we used three feature groups: (a) sentence words, (b) number of new-line characters preceding the sentence (as an approximation of pagination), and (c) the entity tags in the sentence $\mathbf{e}_s$ (for the bottom-up and joint approaches). These features are extracted from the current sentence, the previous two and the following two sentences. Note that we did not tune any of these features in any manner.

4.4. Results and Discussion

Table 3 lists the overall results of the proposed architectures and of three oracle systems. Each oracle system trains only one layer and uses gold information in the other layer during both training and inference, e.g., the claim oracle is a bottom-up system that has access to gold entity labels. The difference between the two entity oracles is that one is fully supervised whereas the other one is semi-supervised, i.e.,

[8] This reduces to a logistic regression model of probability of each variable given its neighbors, in case of undirected exponential models such as ours.

[9] http://nlp.stanford.edu/software/tagger.shtml

[10] The 90 documents came from only 49 cases, so this was an important constraint.

[11] http://nlp.stanford.edu/software/CRF-NER.shtml

Documents	Sentences	Words	Claims	ClaimNumbers	ClaimTypes	Patents	Laws
90	25,250	548,402	362	319	579	1292	433

Table 2: Corpus statistics.

	Claims			Entities		
	Precision	Recall	F_1	Precision	Recall	F_1
Top-down	80.00	54.05	64.52	**86.42**	52.63	65.42
Bottom-up	60.65	50.81	55.29*	48.1	**60.47**	53.58*
Semi-supervised Bottom-up	**89.74**	**56.76**	**69.54***	85.34	56.65	**68.09***
Semi-supervised Joint Hierarchical	88.89	56.22	68.87*	86.16	55.69	67.65*
Claim Oracle	92.40	85.41	88.76	–	–	–
Entity Oracle	–	–	–	83.62	62.99	71.85
Semi-supervised Entity Oracle	–	–	–	85.25	61.77	71.64

Table 3: Overall scores of the proposed architectures and of several oracle models. Asterisks indicate that the difference between the corresponding score and the score of the top-down model is statistically significant. The results for the semi-supervised bottom-up and joint models are not significantly different. All significance tests are performed using two-tailed paired t-test at 95% confidence interval on 20 samples obtained using bootstrap resampling.

the latter trains on $\mathbf{E}^{(\text{semi})}$. We draw several observations from these results:

(a) The performance of the top-down model is reasonable, considering the difficulty of the task and the size and quality of the data. We attribute these results mainly to our hierarchical approach, where each layer models the text at different granularity (sentences or words).

(b) As expected, the first bottom-up approach performs quite badly. This is caused by the skewed entity distribution caused by the partial labeling of the training data, which confuses the claim classifier at inference time.

(c) The semi-supervised bottom-up system addresses this issue successfully. This is our best performing system. This proves that information propagated from the bottom layer improves the top layer significantly. Consequently, the entity layer improves as well, because $\mathbf{E}^{(\text{constraints})}$ (i.e., entities after deleting instances outside claim boundaries) are based on the predictions of the top layer.

(d) The joint model outperforms the top-down model significantly, but it does not perform better than the semi-supervised bottom-up approach. There are two potential causes for this behavior: first, the feedback from the claim model, which has low recall, may end up hurting the performance of the entity layer when computing $P(\mathbf{E}_s^{(\text{semi})}|\mathbf{x}_s, c_s)$; second, because the joint inference must use parallel labels between the two layers, the entity layer self trains on predicted entity labels for data outside of claims, and this may introduce more noise than signal. We can actually quantify the impact of these problems using the two entity oracles. The only difference between the two oracles is that the semi-supervised oracle self-trains its entity model: $P(\mathbf{E}_s^{(\text{semi})}|\mathbf{x}_s, c_s)$ versus $P(\mathbf{E}_s|\mathbf{x}_s, c_s)$. The oracle results indicate that self-training causes a performance drop of .2 F_1 points. Hence, the other .2 F_1 points in the difference between the bottom-up and joint models are caused by the feedback from the claim layer. We conjecture that both these problems are caused by insufficient training data. As more data becomes available, we expect that both self-training the entity layer and the feedback from the claim to the entity layer be successful.

(e) Nevertheless, the table indicates that the joint model improves the precision of the entity layer with respect to the semi-supervised bottom-up model. The entity precision of the joint hierarchical model is .8 points higher than that of the semi-supervised bottom-up model. This is caused again by the feedback from the claim layer to the entity layer. Event though the claim layer in the joint model has low recall, its precision is quite high. This provides precise feedback to the entity layer on where claim boundaries exist, which in turn enhances the precision of the entity layer.

(f) Despite its good performance, the claim oracle actually indicates how difficult this domain is: because gold entities are labeled only inside claims, one would expect this oracle to score close to 100 F_1 points, because any entity mention is a strong hint that the corresponding sentence belongs to a claim. The fact that the claim oracle scores only 88 F_1 points indicates that there is high ambiguity for the sentences not covered by entities.

(g) The relatively low performance of the entity oracles indicates that entity recognition in the legal domain is a hard problem, even when the task is limited at analyzing the text inside claims. We analyze the behavior of our entity models later on this section.

In order to understand the relative importance of various features in the Claim CRF, we perform ablation experiments using the semi-supervised bottom-up architecture. This test involves removing one feature-type at a time and measuring the performance. The results of the test, displayed in Table 4 show that the model is heavily lexicalized – the F1 performance of the CRF drops to as low as 36.02 when words are removed as features. The test also demonstrates that the entities contribute about 5% points in F1, indicating the utility of joint and bottom-up architectures. Surprisingly, pagination does not carry a strong signal for claim identification, and we attribute it to the noisy features resulting from the OCR translation.

Table 5 lists the scores of our best model for each entity type. The table indicates that claim numbers and patents are recognized with acceptable performance, most likely due to their simple structure. In contrast, claim types have low performance. The explanation is that claim type mentions are often complex verbal or nominal phrases, which

	Precision	Recall	F_1
All features	89.74	56.76	69.54
– lexicalization	61.84	25.41	36.02
– pagination	88.33	57.3	69.51
– entities	80.00	54.05	64.52

Table 4: Ablation experiment for the Claim CRF using the semi-supervised bottom-up architecture.

	Precision	Recall	F_1
Claim Number	97.06	54.40	69.72
Claim Type	53.97	26.25	35.32
Law	71.57	36.32	48.18
Patent	94.93	80.94	87.38

Table 5: Results for the entity layer using the semi-supervised bottom-up architecture.

are hard to model using first-order CRFs at word level. We expect more successful models to use full syntax for this entity type. Somewhat surprisingly, mentions of laws are also recognized with low performance. The most common error for this type was caused by the document pre-processor. Law mentions typically include non-ASCII characters (e.g., §), which are mistakenly converted to punctuation marks by the text converters, and these are later seen as end-of-sentence markers by our sentence boundary detector. Since the entity tagger works at sentence level, it cannot recover entities split in different sentences. This is yet another example of a problem that a real-world IE system must address.

For completeness, we show the results of the ablation experiment for Entity CRF in Table 6. To avoid the complex inter-dependencies between the two layers,[12] in this experiment we used the top-down architecture. Similarly to Table 4, this experiment shows that our models are heavily lexicalized: removing lexical features caused a drop in the F_1 score of more than 11 points. The drop is not as high as the drop reported in Table 6 because some of the lexical information is captured by the POS tag and word shape features. The features with the second highest impact are the features extracted from the context surrounding the word to be classified: ignoring this context causes a drop of approximately 3 F_1 points. These observations are consistent with previous work on named entity recognition. What is different in our domain is that POS information does not help when combined with lexicalization: removing POS features yields a slight improvement in the F_1 score. This is caused by the fact that our data is significantly different from the data used to train the POS tagger, both in quality and in domain. Because of this, using the POS tagger in this corpus generates more noise than signal.

Lastly, Figure 4 shows the learning curves for our three best scoring approaches. The curves for Claim CRF show that the bottom-up and the joint systems behave similarly. On the other hand, the top-down approach scores consistently lower, when using more than 20% of the data. For smaller training corpora, the top-down approach performs better

[12]For example, in the bottom-up architecture the claim layer depends on the performance of the entity layer, and, in turn, the output constraints for the entity layer depend on the performance of the claim layer.

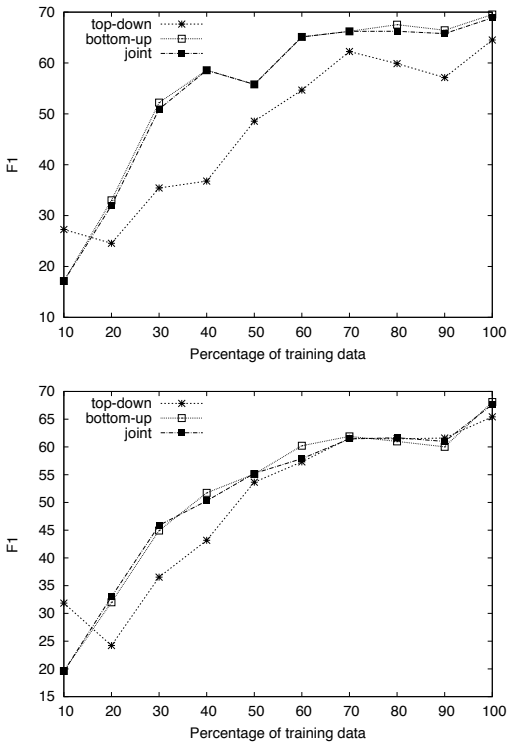

Figure 4: Learning curves of the best three models. The top chart plots the F_1 score of the Claim CRF. The bottom chart plots the F_1 score of the Entity CRF.

	Precision	Recall	F_1
All features	86.42	52.63	65.42
– lexicalization	71.12	43.61	54.07
– POS tags	89.63	51.96	65.79
– word shape	86.80	51.63	64.75
– context	86.32	49.04	62.55

Table 6: Ablation experiment for the Entity CRF using the top-down architecture. "context" indicates all features from the previous and following word. The other three experiments remove the corresponding feature group from all tokens (current, previous, and following word).

because the entity models are not strong enough to provide useful signal in the bottom-up or joint systems. Extrapolating from this observation, we expect that the joint approach will in turn start performing better than the bottom-up one with enough training data. The bottom part of Figure 4 shows a similar story. The differences between the learning curves for Entity CRF are not that large, but they are still statistically significant for the majority of the plot points and they lead to the same conclusions.

5. Related Work

In the field of IE, most body of work –too large to be cited here– falls into one of the two classes described before: flat extractors or deep, semantic extractors. The middle ground has been addressed mainly by works that investigate the recognition of nested named entity mentions, which are common in the medical domain (Alex et al., 2007) and in corpora on languages other than English (Marquez et al., 2007). There are significant differences between our work and nested NER: (a) nested NER is non-hierarchical in the sense that all layers operate at token level, (b) there are no

missing labels in any layer. (Alex et al., 2007) also use a combination of sequential CRF classifiers, but their joint approach focuses on joint representation rather than joint modeling.

The general idea of breaking documents into "zones" with consequences for further processing is not new, e.g., Teufel and Moens used document segmentation based on rhetorical structure for the summarization of scientific articles (Teufel and Moens, 2002). A paper that is closer to ours in terms of using pipelined or joint CRFs for natural language processing from multiple layers is that of (Sutton et al., 2007). In this work, the authors used a two layer factorial CRF to jointly model noun-phrase chunking and POS tagging, and demonstrated significant performance gains compared to a pipelined system of independently trained CRFs. For the same reasons as above, we argue that our problem is more complex than theirs. The work of (McDonald et al., 2007) uses a hierarchical CRF with different levels of granularity (documents and sentences) to model coarse to fine sentiments in a document, but their data is fully observed. Recent work of (Truyen et al., 2008) indeed proposes a hierarchical CRF that incorporates missing labels. They present detailed theoretical treatment of the model in a missing labels scenario, but they test their model only on fully observed data (e.g., joint POS tagging and syntactic chunking).

6. Conclusions

This paper introduces a novel Information Extraction problem, where only parts of documents have relevance and linguistic annotations are available only for these segments. The problem has several hierarchical properties. First, the data is annotated using a two-layer hierarchy: the top layer marks the relevant text segments and the bottom layer annotates domain-specific entity mentions only in these segments. Due to this approach, the data for the bottom layer is only partially labeled, i.e., entity mentions outside of the relevant text segments are not annotated. Second, the two layers are modeled at different granularity: the top layer using the sentence as the atomic element and the bottom layer using words.

We investigate this problem on a real-world application from the IP litigation domain. We introduce two models that outperform significantly the top-down cascaded approach. Using a simple semi-supervised approach for the entity layer we implement a bottom up model and then we extend it to a joint hierarchical CRF. We discuss the advantages and limitations of all approaches.

All in all, this work shows that complex IE systems can be built and trained using hierarchical, partially-labeled data. We believe that this reduces annotation efforts, which is an important constraint in the development of any supervised IE system. To further improve the performance of our system without increasing the annotation burden on the legal experts we plan to: (a) combine our approach with unsupervised topic segmentation algorithms (Allen, 2002), which will be used to enhance our claim extractor, and (b) combine our models with rule-based systems, e.g., we expect a rule-based patent mention extractor to perform well, and to provide hints about where claim information is concentrated. On the legal side of project, in future work we will extend our entity extraction model with other entity types of interest, e.g., product names, and our claim detection model with other types of claims, e.g, trade secret or trademark violation.

Acknowledgments

We thank the Stanford Intellectual Property Litigation Clearinghouse (IPLC) project[13] for their annotation effort and Jenny Finkel for the help with her CRF implementation.

7. References

B. Alex, B. Haddow, and C. Grover. 2007. Recognising nested named entities in biomedical text. In *Proc. of BioNLP 2007*.

J. Allen. 2002. Introduction to topic detection and tracking. *Topic Detection and Tracking: Event-Based Information Organization*.

C. Andrieu, N. De Freitas, A. Doucet, and M.I. Jordan. 2003. An introduction to MCMC for machine learning. *Machine Learning*.

J. Besag. 1975. Statistical analysis of non-lattice data. *The Statistician*, 24:179–195.

J. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems through Gibbs sampling. In *Proc. of ACL*.

D. Freitag. 1998. Machine learning for information extraction in informal domains. *Ph.D. thesis, Carnegie Mellon University*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*.

L. Marquez, L. Villarejo, M.A. Marti, and M. Taule. 2007. SemEval-2007 task 09: Multilevel semantic annotation of catalan and spanish. In *Proc. of SemEval-2007*.

R. McDonald, T. Neylon K. Hannan, M. Wells, and J. Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *Proc. of ACL*.

S. Parise and M. Welling. 2005. Learning in Markov Random Fields: an empirical study. In *Joint Statistical Meeting*.

E.F. Tjong Kim Sang and F. De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*.

C. Sutton, A. McCallum, and K. Rohanimanesh. 2007. Dynamic Conditional Random Fields: Factorized probabilistic models for labeling and segmenting. *The Journal of Machine Learning Research*.

S. Teufel and M. Moens. 2002. Summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4).

K. Toutanova, D. Klein, C. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of HLT-NAACL*.

T.T. Truyen, D.Q. Phung, H.H.Bui, and S. Venkatesh. 2008. Hierarchical Semi-Markov Conditional Random Fields for recursive sequential data. In *Proc. of NIPS*.

[13] http://www.law.stanford.edu/program/centers/iplc/

On the Extraction of Decisions and Contributions from Summaries of French Legal IT Contract Cases

Manuel Maarek

LICIT
INRIA Grenoble Rhône-Alpes
Manuel.Maarek@inrialpes.fr

Abstract

French court decisions play an ambivalent role of being the main source for software contract drafters and in the meantime an unreliable legal authority as precedents are not creating laws in the French legal system. Nevertheless, in the legally yet unsettled domain of IT contracts, case decisions are the main source of legal information for contractors. Thus, semantic extraction from French legal IT contract cases could highlight valuable information. We have experimented the extraction of decisions and contributions from a set of summaries of French legal IT contract cases. We are reporting on the ongoing development of our method to represent and infer such semantic knowledge from legal summaries.

1. Introduction

Legal cases are the concrete realisation of the law in a real context. Under *Common Law* systems, rulings by judges in court are a prominent source of law. An important aspect of the work of lawyers is to gather, interpret and reuse relevant precedents to strengthen their own argumentation or ground their decisions.

The processing of legal cases in Common Law systems have found great interest in the Law and IT (Information Technology) research community. The goals and uses of such computations are numerous, ranging from the indexing (Klein et al., 2006) and abstracting (Uyttendaele et al., 1998) of cases to their formalisation and inclusion in argumentation frameworks (Wyner et al., 2009).

In contrary to Common Law systems, the French legal system does not usually consider cases as a source of law. Precedents are not creating laws and judges should refer only to statutes in their decisions.[1] Precedents are therefore less explicit in the French legal system than in other legal systems. In the French legal system, precedents become meaningful when they reveal a strong tendency of similar judgements. Some decisions (*arrêts de principe*) by the *Cour de Cassation* are considered to set a precedent, and therefore to orientate the practice of judges, as they attempt to resolve a controversial and purely legal question. The studies of the law and legal cases (*Doctrine*) can reveal tendencies and influence the law makers to create new statute laws and therefore reduce legal uncertainty.

The relative novelty of IT has created a gap between legal practice and legal regulation. This results in a situation of legal uncertainty. In this respect, inferring tendencies from court decisions on IT related cases is of high importance as a support for parties and lawyers, and ultimately for shaping future software laws.

1.1. Motivations and goals

Considering both the novelty of legal issues in IT and the specific situation of cases in the French legal system, a computer-aided extraction of semantic information from past cases in IT contracts opens up new possibilities:

- A refined indexing and sorting of IT contract cases which not only relies on date and jurisdiction but also on context, factors and kind of decisions.
- A mapping of related cases (according to the decision and the factors implied in the decision) which automates the task of retrieving common decisions and therefore tendencies in court decisions.

1.2. LISE

The work we present here is taking place in LISE,[2] a multidisciplinary project involving lawyers and computer scientists. We aim at defining in a precise way liability in software engineering (Le Métayer et al., 2010). To give a broad perspective on the legal practices in software liability, we have studied court decisions related to IT contract litigations (Hardouin, 2009). As a basis for this study, Ronan Hardouin and Sylvain Steer from the DANTE laboratory[3] have produced a table containing information about summaries of IT contact legal cases. They have presented an analysis of the overall table in (Hardouin, 2009). We present in this paper preliminary works on the extraction of semantic information from the case summaries of this table.

1.3. Approach and contributions

The starting point of our work (Section 2.) is a table of case summaries. Each of these summarise highlights the court decision itself and the contributing factors on which the case decision is based. We propose a semantic representation of court decision and contributions (Section 3.) and present our approach for extracting such semantic knowledge from these summaries (Section 4.). We later detail the

[1] Article 5 of the *Code Civil* enacts the separation of powers between the legislature which creates the law and the judiciary which applies it.

[2] LISE (Liability Issues in Software Engineering) is a project funded by ANR (Agence Nationale de la Recherche) under the SeSur 2007 programme (ANR-07-SESU-007). http://licit.inrialpes.fr/lise/

[3] DANTE (*Droit des Affaires et Nouvelles Technologies*), partner in the LISE project, is a law research laboratory at the University of Versailles Saint-Quentin-en-Yvelines.

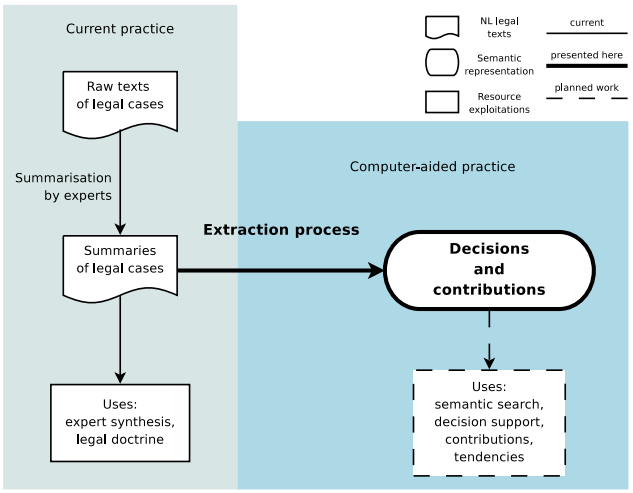

Figure 1: Approach

prospective usages we are developing or intend to develop in future works (Section 5.). Figure 1 sums up in a diagram our overall approach.

2. Resource

As we mentioned in Section 1.2., the starting point of this work is an overview and study, by the DANTE laboratory, of French legal cases related to IT contracts. We present in this section the context in which this study took place and the kind of information gathered in the DANTE table of legal cases.

2.1. Context: French IT contract related cases

The context in which the DANTE table of IT contract cases was produced is certainly transitory for IT French law. In the current situation, some important issues related to IT contracts, such as disclaimer of liability and indivisibility of contracts, do not get harmonised treatments in court. This is mainly due to the novelty of IT and the frequent changes of practices in the IT domain combined with a slow legislative process.

2.2. Data: cases and summaries

The version of the DANTE table we use here focuses on cases dating from 2000 to 2009. For each entry of the table, the meta information is isolated: the decision date, the appealed decision, the parties involved in the case, the case identification number, the topic of the case[4], the legal founding articles if any, the solution of the case, the contribution of the case and existing published comments on the case. Table 1, recalls this description of the table columns. In this paper, we focus on the solution and contribution columns which contain natural language summaries of each case. In the summaries, only the essential obligations and the pertinent factors (*moyens*) are recalled.
Table 2 lists the prominent topic (themes) attributed to the entries in the DANTE table. The main topics treated by the set of cases concern contractual liability, especially the duty to advise, delivery in conformance with the contract, contract annulment and contract termination.

[4]Note that a case addressing several issues would have several entries in the table, each entry targeting one topic.

3. Representing judicial decision and contribution

Our objective being to group and search for similar judicial cases according to the relevant grounds stated by the parties, we propose a general representation for decisions and contributions.

3.1. Context

Before we describe the manner we represent a judicial decision and its contributions, we need to define the ingredients that will compose such representation. These ingredients are of three kinds describing a contract, an obligation, or a pertinent factor brought to the court.

- The *nature of the contract* are within the limits of the computer context: software or hardware supply, or IT service provider.
- The *obligations* are the duties imposed either by contract, by law or by principle. They are considered by the plaintiff not fulfilled by the defendant. Moreover, the plaintiff usually suffered losses or damages from default on an obligation, and has arguments proving it.
- The *factors* brought to the court by the parties are either concrete facts or considerations. Facts are either mentioned as having occurred or as being missing.

We denote a nature of contract by $[expression]$, an obligation by $\odot expression$, the presence of a factor by $+expression$, and the absence of a factor by $-expression$. Here *expression* is respectively the unique naming of a nature of contract, an obligation and a factor.

3.2. Decisions

In a contractual litigation, the final judgement concludes, in general, either in the fulfilment of an obligation or the default on an obligation by one of the parties. We represent this general dichotomy for an obligation $\odot expression$

Column name	Description
Jurisdiction	Type of court or tribunal
Date	From 2000 to 2009
Appealed decision	Jurisdiction and date of the appealed decision
Parties	Plaintiff and defendant
Reference	Index entry in jurisdiction database
Themes	Classification with three level of attributed subjects
Foundation	Main articles on which the decision is based
Solution	Outcome of the legal decision
Contribution	Solution and contribution brought by the decision
Comments	Comments coming from doctrinal articles

Table 1: Information for each legal case

Entries	145
Cases	128
Appeals Only 7 out of the 128 decisions treated here are initial decisions because most of the initial decisions are not automatically published by the jurisdictions.	121
Themes (57)	
Major themes (3)	
Droit d'auteur (copyright)	4
Responsabilité contractuelle (contractual liability)	137
Responsabilité délictuelle (torts)	4
Sub-themes (21)	
Devoir de conseil (duty to advise)	29
Délivrance conforme (delivery in conformance with the contract)	27
Résolution (contract annulment)	20
Résiliation (contract termination)	14
Ensemble contractuel (contract entirety)	11
Préjudice (damage)	10
Sub-sub-themes (33)	
Conditions (conditions)	38
Indivisibilité (contractual indivisibility)	11

Table 2: Spreading of the table's case entries

by $\oplus$*expression* for a fulfilment of the obligation, and by $\ominus$*expression* for a default on the obligation.

3.3. Contributions

A court decision is grounded on factors which were argued and discussed by both parties. In its decision, the court recalls the factors that made his decision. In the summaries of the DANTE table, only the pertinent and significant factors are recalled. We represent the contribution of a decision by putting in relation these pertinent factors with the decision they participated to.

$$factors \rightsquigarrow decision$$

This expression reads "the *factors* participated to the *decision*". Note that the set factors on the right side is not exhaustive. Some factors are omitted by the summarisation process to reveal only pertinent ones.

The outcome of a new case decision could be positioned within the set of related cases depending on its novelty with former case or its replication of a former case. The ruling of a new case can either confirm former rulings, contradict it or extend the factors use to show a fulfilment of or a default on an obligation. In the situation of our set of cases, the contributions of each case has been highlighted in the summaries table. The judicial summarisation of the case decision by DANTE lawyers was intended to identify the relevant and determining factors used by the parties as well as the novelty of the court decision.

We propose to represent contributions with four constructs.

Attack To convince the judge that the opposite side did not fulfil its obligation o, a party made use of factor f and won the case. The outcome is that this obligation was attacked by this factor:

$$f \rightsquigarrow \ominus o$$

Defence To convince the judge that the opposite side did not fulfil its obligation o, a party made use of factor f but lost the case. The outcome is that this obligation rebutted this factor:

$$f \not\rightsquigarrow \ominus o$$

Consolidation To convince the judge of the fulfilment of the obligation o, a party made use of factor f and won the case. The outcome is that this factor consolidates or backs the fulfilment of the obligation.

$$f \rightsquigarrow \oplus o$$

Restraint To convince the judge of the fulfilment of the obligation o, a party made use of factor f but has lost the case. The outcome is that this factor was not accepted for the fulfilment of this obligation.

$$f \not\rightsquigarrow \oplus o$$

4. Extracting judicial decision and contribution

We develop our method for extracting, from summaries of French legal IT cases, the legal decision and the contribution of each case (in terms defined in Section 3.). We assume here, that the task, performed by lawyers, of summarising the cases has identified the pertinent elements of the judicial case and highlighted the novelty and significance of each case. We first explain our extracting method (Section 4.1.) and then go through an extraction example (Section 4.4.).

4.1. Extraction method

The method we used is decomposed in three steps.

1. *Sequencing text.* We first decompose the text into a sequence of words.
2. *Identification of sequences.* We then search in the full sequence of words for specific sub-sequence matching a corpus of pre-defined sequences. The sequences being identified could be intermixed (several sub-sequences could use the same words from the full sequence). The corpus of pre-defined sequences (see Section 4.2.) are sorted by categories depending on their role in the text (see Section 4.3.).
3. *Recognition of narrative structures.* We then analyses the ordering of sub-sequences to infer the legal decision and contributions of the case. We use pre-defined narrative schemes for this recognition (see Section 4.2.).

We validate the result of an extraction by verifying that the output is a set of well formed decisions and contributions. Note that we do not use Natural Language Processing (NLP) per se, nor use any statistical method as used for instance in (Lame, 2004). Therefore, the sequence of words we are willing to identify should be meaningful enough for their appearance to be directly interpreted.

4.2. Ad-hoc corpus incrementally defined

The texts we analysed are specific in several senses. They use both IT and legal vocabularies, and they are usually composed of expressions and sentences originating directly from the court's legal decisions itself. Due to these specificities, the textual structure of the summaries is very peculiar and not standard which therefore makes the use an ad-hoc corpus more appropriate for processing it. Nevertheless these peculiarities are homogeneous throughout the summaries. The use of an ad-hoc corpus is also more adequate for a relatively small set of cases (145 summaries made of an average of 34.4 words each).

We have built our set of word sequences and our set of narrative schemes in an incremental manner. For each case summary, we identify the important sub-sequence of words, we name and categorise them according to their role in the text (see Section 4.3.). This naming is specific to the set of cases we are dealing with. We then define a rule identifying the narrative structure of this summary. These sets of sequences and rules are then used by our text analysing engine for extracting the decision and contributions out of the summary. The generic representation of decisions and contributions we presented in Section 3. gives us the possibility to check the well formation of the extracted knowledge. The process is incremental as it makes use of the definitions of sequences and schemes for prior case summaries to facilitate and accelerate the extraction in later ones. Sequences and schemes are usable throughout the summaries thanks to the precision of the judicial texts we analyse.

4.3. Categories of word sequences

We use five categories of word sequences. The three first are composed by the ingredients of the summaries (Section 3.1.), the two remaining ones by the elements participating in its narration.

$\mathcal{N}$	Nature of the contract.
$\mathcal{O}$	Obligation by contract, by law or by principle.
$\mathcal{C}$	Factor (fact or consideration) used to ground the decision.
$\mathcal{A}$	Elements giving the structure of the argumentation.
$\mathcal{M}$	Modifiers.

The narrative schemes we use to extract decisions and contributions are defined using these categories and the names of word sequences we have identified. For instance, in the example that follows, we name the sequence of words "*ne permet pas*" ("does not allow") as a refutation which is of category $\mathcal{A}$ as it contributes to the structure of the argumentation of the sentence. Similarly, both word sequences "*responsabilité du vendeur*" (supplier's responsibility) and "*proposition de matériel*" (hardware offer) inform of the nature of the contract (a hardware supply[5]) and therefore were attributed with the same name (hardware supply) and category ($\mathcal{N}$). At this stage we did not use a pre-defined ontology for these names and categories.

[5] In our specific set of cases, all supplies are hardware supplies.

4.4. Example

1. Let us go through an example of the extraction of information from a case summary. The case in question is a February 2009 ruling by the *Cour d'Appel de Dijon*. The summary is as follows.

Case summary
$^{\mathcal{M}}$*<L'absence de> possibilité d'établir un* $^{\mathcal{C}}$*<cahier des charges> précis et détaillé et la* $^{\mathcal{C}}$*<modification continuelle des besoins> du client au cours de l'exécution d'un contrat d'installation d'un ensemble informatique personnalisé* $^{\mathcal{A}}$*<ne permet pas> d'exonérer la* $^{\mathcal{N}}$*<responsabilité du vendeur> qui doit imposer ou réaliser une* $^{\mathcal{O}}$*<analyse fonctionnelle des besoins> avant toute* $^{\mathcal{N}}$*<proposition de matériel>.*

2. We have identified a number of word sequences that are highlighted in the text. We list these sequences here with their categorisation and their interpretation both in French and English.

$\mathcal{M}$	*Absence*	$=$	m_1
	Lack		
$\mathcal{C}$	*Cahier des charges*	$=$	c_1
	Specifications		
$\mathcal{C}$	*Modification des besoins*	$=$	c_2
	Change of needs		
$\mathcal{A}$	*Réfutation*	$=$	a_1
	Refutation		
$\mathcal{N}$	*Fourniture materiel*	$=$	n_1
	Hardware supply		
$\mathcal{O}$	*Analyse besoins*	$=$	o_1
	Analysis of needs		
$\mathcal{N}$	*Fourniture materiel*	$=$	n_2
	Hardware supply		

3. The next step is to extract decisions and contributions from this list of items.

$$m_1\ ;\ c_1\ ;\ c_2\ ;\ a_1\ ;\ n_1\ ;\ o_1\ ;\ n_2$$

We first isolate the items informing on the nature of the contract. In this example, we have two items n_1 and n_2. We deduce that the nature of the contract is a hardware supply as $n_1 = n_2 = [$*Hardware supply*$]$. We now analyse the rest of the items.

$$m_1\ ;\ c_1\ ;\ c_2\ ;\ a_1\ ;\ o_1$$

We focus on the factors and modifiers to differentiate absent and present factors. We use a simple pattern here for factors which transforms sequences of the form $^{\mathcal{M}}$*Lack* ; $^{\mathcal{C}}c$ into absent factors, and other $\mathcal{C}$ items into present factors.

$$-c_1\ ;\quad +c_2\ ;\ a_1\ ;\ o_1$$

We then use narrative schemes to identify the contribution. In this example, the narrative scheme is common in the DANTE table and has the form:

c ; ... ; c ; $^{\mathcal{A}}$*Refutation* ; o which we transform into a defence: $c, \ldots, c \not\rightsquigarrow \ominus o$. The result is the following contribution:

$$-c_1, +c_2 \not\rightsquigarrow \ominus o_1$$

The corollary of this contribution is the actual decision of the case which is a default to obligation o_1 in $[n_1]$.

$$\ominus o_1[n_1]$$

The result of the extraction is these well formed decision and contribution. We inlined the expressions of the two formulas and give a rough translation in natural language.

Decision
$\ominus$*Analysis of needs*[*Hardware supply*]
Default on the obligation of analysis of needs in hardware supply.

Contribution
$-$*Specifications*, $+$*Change of needs* $\not\rightsquigarrow \ominus$*Analysis of needs*
Defence of the obligation of analysis of needs against the absence of specifications and the presence of change of needs.

This extraction process obviously misses, from the original text, some information that one could consider important but makes some automation possible. As we mentioned, the refutation scheme and the absence transformation get very often repeated throughout the summaries table.

5. Follow-ups and future works

In this section we sketch the ongoing works which immediately follow this task of extracting decisions and contributions. We also mention some future works we envision.

5.1. Follow-ups

In this paper, we have intentionally left out discussions on the evaluation and on the uses of the results of such extraction to focus on explaining our representation for case decisions and contributions, and on describing the method we have used to extract such information from case summaries.

Evaluation. The quality of the extraction depends on the corpus that gets incrementally built by hand during the process. Comparing the results of our extraction with the rest of the information already provided in the table (comparison with the decision and contribution of the appealed decisions, nature of the parties and concordance of the extraction on the solution and contribution columns) is an evaluation that we have intended to perform.

Compiling results. The extraction results in a set of decisions and contributions. The compilation of these results expressed with our semantic representation gives much possibilities of semantic reasoning. We could compile and compare the results to draw a map of the tendencies in legal decisions following three axes: (1) an *obligation-factor* relation which links the factors implied in the argumentation pro and cons an obligation, (2) a *factor-factor* graph of competing factors, and (3) a *jurisdiction-contribution* relation highlighting tendencies of a particular jurisdiction to rule in a certain direction.

Uses. In the LISE project, we aim at providing computer-aided methods for assisting decision at different stages of the judicial process: (1) *contract drafting*: prior to the signature of a contract, a party could gain help from such knowledge on former decisions to estimate the risk and shape clauses accordingly, (2) *dispute*: during an out-of-court settlement or a court litigation, the parties would reach conciliation or, in the case of a trial, refine their arguments for a dispute based on similar precedents retrieved by this extraction, and (3) *doctrinal study*: the evaluation of the regularity and homogeneity of past regulations could be of great help for law makers.

5.2. Future works

We envision some further extensions of this work.

NLP enhancement.
As we mention in Section 4.1., we did not employ NLP per se in this work but we understand that NLP techniques would enhance the extraction process and widen the range of reusability and generality of the corpus of word sequences. For instance, the identification of the absence (of our category $\mathcal{M}$) resembles NLP negative analysis.

Computer-aided corpus creation. The method we presented relies on the creation of corpuses of word sequences and narrative schemes. This creation needs to be performed by lawyers to be qualitative. To smoothen this task, it is important to provide user-friendly tools. In this respect, a corpus-creation by demonstration similar to the manual annotation and validation system proposed in (Kamareddine et al., 2007) would certainly be suitable. Alternatively, the possibilities offered by Controlled Natural Languages (CNL, 2009) could also be applicable.

Extraction from court decisions. In this paper, we presented an extraction of information from summaries made by lawyers. Comparing such extracted knowledge with extractions made directly from entire court decisions, as presented in (Stede and Kuhn, 2009), would be of interest but would require a larger representation model for court argumentation. This work would also be a starting point for automatic summarisation as presented in (Chieze et al., 2008).

6. Related Work

There exists several methods for processing and extracting information from legal texts. Each method suits better particular motivations and needs which depend on the legal corpus and the targeted application. We list a selection of works from this field of research which are of interest for this work and its continuation.

Legal case-based reasoning. The follow-up study of compiling the extracted knowledge fall into the domain of legal case-based reasoning. The analysis presented in (Wyner and Bench-Capon, 2007) which combines legal cases in terms of argument schemes, is certainly applicable to our set of cases.

Legal argumentation scheme extraction. The extraction of tendencies from a body of cases which do not explicitly refer to each others necessitates a formal semantic representation of the extracted knowledge. Recent works (Mochales Palau and Moens, 2008; Mochales Palau and Moens, 2009; Wyner et al., 2009) orient text-mining and NLP techniques for the extraction of argumentation schemes and question the composition of the semantical results of the extraction.

Knowledge extraction from legal cases. In the specific issue of legal cases matching, (Klein et al., 2006) proposed a methodology which uses a user description of its own case situation for retrieving similar former legal cases. This methodology makes use of a lightweight semantic processing of the legal cases and of ontology matching. They later reported in (Hoekstra, 2009) on their experiment and moved to the development of processing with weightier semantics.
The extraction of information from legal cases is often done by searching for patterns of sentences that get replicated in legal discourse. For example, (Chieze et al., 2008) uses such method for summarising legal cases.

Structure of legal argumentation. The representation of decisions and contributions we presented in this paper is one small aspect of research in legal argumentation. The formalisation of legal argumentation ranges from adaptations to the legal domain of the Toulmin model of argument (Toulmin, 1958) to recent works on legal argumentation frameworks (Prakken, 2009).

7. Conclusion

We have presented an experiment on the extraction of decisions and contributions from summaries of French legal IT contract cases. We outlined the original settings of dealing with decisions in the French legal system which is Statute Law, and of focusing on the unsettled legal domain of IT contracts. To pursue this task, we have designed a model for representing legal decisions and contributions and we have develop a method for extracting such knowledge from natural language summaries of court decisions.

8. Acknowledgements

We would like to acknowledge and thank Ronan Hardouin and Sylvain Steer from DANTE Laboratory. This work has been funded by ANR under the SeSur 2007 programme (ANR-07-SESU-007, LISE project).

9. References

Emmanuel Chieze, Atefeh Farzindar, and Guy Lapalme. 2008. Automatic summarization and information extraction from canadian immigration decisions. In *Proceedings of the Semantic Processing of Legal Texts Workshop*, pages 51–57. LREC 2008.

CNL. 2009. *Workshop on Controlled Natural Language (CNL 2009)*, Marettimo Island, Italy, June 8–10.

Ronan Hardouin. 2009. Le sens des responsabilitiés en matière de contrats informatiques. Technical report, Livrable LISE D1.1.

Rinke Hoekstra. 2009. BestPortal: Lessons learned in lightweight semantic access to court proceedings. In *Proceedings of the 22nd International Conference on Legal Knowledge and Information Systems (JURIX 2009)*. IOS Press.

Fairouz Kamareddine, Robert Lamar, Manuel Maarek, and J. B. Wells. 2007. Restoring natural language as a computerised mathematics input method. In *Towards Mechanized Mathematical Assistants (Calculemus 2007 and MKM 2007 Joint Proceedings)*, volume 4573 of *LNAI*, pages 280–295. Springer-Verlag.

Michel C. A. Klein, Wouter Van Steenbergen, Elisabeth M. Uijttenbroek, Arno R. Lodder, and Frank van Harmelen. 2006. Thesaurus-based retrieval of case law. In *Legal Knowledge and Information Systems. JURIX 2006: The Nineteenth Annual Conference*, pages 61–70. IOS Press.

Guiraude Lame. 2004. Using NLP techniques to identify legal ontology components: Concepts and relations. *Artif. Intell. Law*, 12(4):379–396.

Daniel Le Métayer, Manuel Maarek, Eduardo Mazza, Marie-Laure Potet, Stéphane Frénot, Valérie Viet Triem Tong, Nicolas Craipeau, Ronan Hardouin, Christophe Alleaume, Valérie-Laure Benabou, Denis Beras, Christophe Bidan, Gregor Goessler, Julien Le Clainche, Ludovic Mé, and Sylvain Steer. 2010. Liability in software engineering – Overview of the LISE approach and illustration on a case study. In *32nd International Conference on Software Engineering, ICSE 2010, Proceedings*. ACM. To appear.

Raquel Mochales Palau and Marie-Francine Moens. 2008. Study on the structure of argumentation in case law. In *Legal Knowledge and Information Systems, JURIX 2008: The Twenty-First Annual Conference*, pages 11–20. IOS Press.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: the detection, classification and structure of arguments in text. In *ICAIL*, pages 98–107.

Henry Prakken. 2009. An abstract framework for argumentation with structured arguments. Technical Report UU-CS-2009-019, Department of Information and Computing Sciences, Utrecht University.

Manfred Stede and Florian Kuhn. 2009. Identifying the content zones of german court decisions. In *Business Information Systems Workshops, BIS 2009 International Workshops, 2009.*, volume 37 of *Lecture Notes in Business Information Processing*, pages 310–315. Springer.

Stephen Edelston Toulmin. 1958. *The Uses of Argument*. Cambridge University Press.

Caroline Uyttendaele, Marie-Francine Moens, and Jos Dumortier. 1998. Salomon: Automatic abstracting of legal cases for effective access to court decisions. *Artif. Intell. Law*, 6(1):59–79.

Adam Zachary Wyner and Trevor J. M. Bench-Capon. 2007. Argument schemes for legal case-based reasoning. In *Legal Knowledge and Information Systems. JURIX 2007*, pages 139–149. IOS Press.

Adam Wyner, Rachel Mochales Palau, Marie-Francine Moens, and David Milward. 2009. Approaches to text mining arguments from legal cases. To appear in LNCS.

Towards Annotating and Extracting Textual Legal Case Factors

Adam Wyner[1], **Wim Peters**[2]

[1]University College London, [2]University of Sheffield
adam@wyner.info, w.peters@dcs.shef.ac.uk

Abstract

Case based reasoning is a crucial aspect of *common law* practice, where lawyers select precedent cases which they use to argue for or against a decision in a current case. To select the precedents, the relevant facts (the case factors) of precedent cases must be identified; the factors predispose the case decision for one side or the other. As the factors of cases are linguistically expressed, it is useful to provide a means to automate the identification of candidate passages. We outline and report the results of our approach to the identification of legal case factors which follows a bottom-up knowledge heavy strategy and uses the General Architecture for Text Engineering system. Salient lexical items are selected, concept classes of related terms are created, and annotation rules for simple and compound concepts are provided. The annotated concepts can be extracted from the cases, and cases can be classified with respect to the concepts. In addition to supporting extraction of relevant information, the approach has a didactic use in helping to train lawyers to perform close textual analysis. Finally, we carry out an initial collaborative, online annotation exercise using GATE TeamWare in order to develop a gold standard.

1. Introduction

Case based reasoning is a crucial aspect of *common law*, where lawyers argue a current undecided case on the basis of legal precedents, which are decided cases drawn from a legal case base.[1] The lawyers compare and contrast the current undecided case against decided cases in terms of the facts of the cases and the applicable laws. Based on the facts and arguments, judges and juries decide a case, guided by a conservative principle of *stare decisis*, which obliges the decision to be consistent with decisions of previous cases *ceteris paribus* (though sometimes decisions are overturned); where the facts of the cases vary, the lawyers and judges reason with respect to a counterbalancing of factors and their role in the law and society. Prototypical fact patterns are referred to as *factors* and the analysis of the factors in a case is *factor analysis*; a given factor may predispose the case to be decided in favour of one side or the other of the dispute. For instance, in the domain of *intellectual property* cases, where a plaintiff claims a defendant stole the plaintiff's intellectual property, a factor would be whether or not the plaintiff required the defendant to sign a non-disclosure agreement prior to disclosing the secret. If the plaintiff did not require the agreement, this fact would predispose the decision in the case in favour of the defendant since, after all, the lack of a requirement indicates that the plaintiff was negligent in identifying and protecting his property. On the other hand, if the plaintiff did require the agreement, but the defendant did not abide by it, then this fact would predispose the decision in favour of the plaintiff since the plaintiff was making efforts to protect his property, but the defendant violated the agreement. The actual outcome of the case depends on the full range of factors, their relationships, the law, and procedural moves by the lawyers, among other aspects that contribute to a court decision. Thus, it is crucial to determine what factors hold of a case as reported in the language of the case decision (which is distinct from determining the facts in the first instance), both for research and in practice.

The goal of this paper is to demonstrate the feasibility of applying automated tools to support the identification of factors and to annotate them for subsequent information extraction and processing. Text annotation in general and factor annotation in particular of unstructured linguistic information is a complex, time-consuming, error-prone, and knowledge intensive task; it is a difficult aspect of the "knowledge acquisition bottleneck" in information processing (Forsythe and Buchanan, 1993). Techniques which facilitate factor analysis would help lawyers find relevant cases. In addition, by using Semantic Web technologies such as XML and ontologies, novel methods could be developed to analyse the law, make it more available to the general public, and to support automated reasoning. Nonetheless, the development of such technologies depends on making legal cases structured and informative for machine processing.

In this work, the semantic annotations are the leaves of a hierarchy of factors, where the leaves indicate higher level factors to a lesser or greater extent, rather than precisely. In general, factors constitute conceptual entities in legal discourse, which can be of various levels of semantic complexity. At the lowest level, factors can be regarded as similar to linguistic expressions such as domain-specific nominal, adjectival, or verbal terms and keywords. These combine into increasingly complex higher level factors such as collocations or verbal predicates. The workflow we are aiming at accommodates all factor levels by annotating higher level factors in terms of lower level constituents. It allows an incremental bridging of levels by means of the addition of fine-grained domain-specific patterns of language use, in whichever linguistic form. In order to provide the initial linguistic building blocks to bridge levels of linguistic description, we apply text preprocessing steps such as sentence splitting, tokenisation, part-of-speech tagging, and lemmatisation. The flexible combination of these building blocks makes possible the mapping of the surface language onto the underlying conceptual factor organization of the legal case domain. This initial study will lead to further research, where we will iteratively refine the annotations to more closely approximate the linguistic realisations.

 Corresponding author: Adam Wyner, adam@wyner.info.

In this paper, we apply natural language information extraction techniques to a sample body of cases, which are unstructured text, in order to automatically identify and annotate the factors. Annotated factors can then be extracted for further processing. Not only does such an approach offer to save time and money, but it also reveals key elements at the basis of legal case based reasoning and advances research in AI and Law. In section 2., we first outline some background and the materials. In section 3., we detail the methodology, which uses the General Architecture for Text Engineering(GATE) system, and the results of our method. In section 4., we outline a manual annotation experiment using GATE TeamWare, which allows comparison to the automatic annotation, and the results of the experiment. In section 5., we compare our approach to the key previous approach to factor extraction. Finally, in section 6., we summarise our report and outline future work to improve our results. Overall, we demonstrate the feasibility of our approach and the opportunities for open source, collaborative refinement.[2]

2. Background and materials

2.1. Background

Legal case based reasoning with factors has long been a research area in AI and Law. For our purposes, we can identify two main branches of research. One branch develops knowledge representations of cases and reasoning systems over a knowledge base. While the knowledge base may be derived from a textual case base, most often this is done by manual analysis, where the knowledge representation abstracts from the text and the reasoning rules apply to the abstract elements of the knowledge base (cf. (Hafner, 1987), (Ashley, 1990), (Rissland et al., 1996), (Aleven, 1997), (Chorley, 2007), (Rissland et al., 2006), (Wyner and Bench-Capon, 2007), (Wyner, 2008)). However, this line of research does not address the knowledge bottleneck.

The other branch attempts to address the bottleneck with textual analysis – the annotation and extraction of information from its linguistic realisation – using NLP techniques for ontology construction ((Lame, 2004), (Maynard et al., 2008), and (Peters, 2009)), text summarisation ((Moens et al., 1997) and (Hachey and Grover, 2006)), and extraction of precedent links (Jackson et al., 2003). However, these are tangential to our topic. Somewhat more relevant is (Maxwell et al., 2009), where events are extracted using part-of-speech tags, heads of arguments of predicates, and syntactic dependency structures; such a technique might be applicable to the identification of some factors, though that is not the object of study in (Maxwell et al., 2009).

While factor analysis and factor reasoning is of long practice in the law, formal, automated approaches are relatively more recent ((Ashley, 1990) and (Aleven, 1997)). In the CATO system of (Aleven, 1997), a case base is manually analysed, and factors are associated with the cases. CATO provides as well automated means to support reasoning about the cases with respect to the cases in order to propose a decision. Current versions of CATO provide a system for students to index cases and argue about them (Aleven, 2003).

Figure 1 is an example from (Aleven, 1997) where students are presented with a case, *Mason v. Jack Daniel Distillery*, a list of potential factors such as *Security Measures* and *Unique Product* among others, and guidance on how to identify the factors in the text. When the factor is identified, a note is made alongside the text.

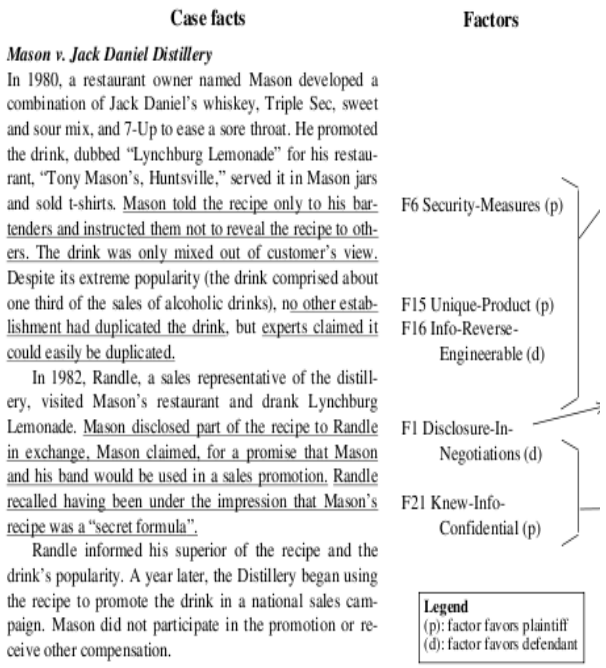

Figure 1: A case with associated factors

While the textual elements are associated with the factors, and cases are thereby indexed with respect to the factors, the association is manual and the result is not an annotation since the factor note does not mark the text directly.

However, just what constitutes a factor is not formally defined, but informally given as a description along with indications of when the factor does and does not hold. One of the questions our research highlights is the structure of factors – are they schemes, events, or frames? Moreover, just what is the relationship between the lowest level linguistic indicators and higher level compound concepts?

2.2. Materials

For materials, we have drawn from the CATO corpus of cases and the CATO factors in (Aleven, 1997). Our reason is that this is a narrowly defined set of cases and factors; moreover, it is a well-studied and well-developed domain, so integrating our presentation in the context of ongoing work. As such, we can leverage the previous results and compare our results to them. Furthermore, by gathering and annotating the cases, the CATO case base can be made available to a wider range of researchers for experimentation.

The CATO corpus is comprised of some 140 cases concerning intellectual property. However, all legal case decisions

[2] All the materials, lists, and JAPE rules are available for testing and development under an Attribution-Non-Commercial-Share Alike 2.0 license. Contact the first author for the files.

are not yet openly or freely available. Of the 140, we have gathered 39 which are available. Of these, we have selected four to work with in order to narrow the scope of the current project; we discuss the rationale for our selection below. These cases are:

- FMC Corp. v. Taiwan Tainan Giant Ind. Co, Ltd, 730 F.2d 61 (2nd Cir.1984) (FMC)
- Goldberg v. Medtronic, 686 F.2d 1219 (7th Cir. 1982) (Gold.)
- Midland-Ross Corp. v. Yokana, 293 F.2d 411 (3rd Cir.1961) (Mid.)
- Trandes Corp. v. Guy F. Atkinson Co., 996 F.2d 655 (4th Cir.1993) (Tra.)

(Aleven, 1997) discusses 27 *base level factors*, which are distinct from the intermediate and higher level factors. Factors are associated with the side of the case that they support, either the plaintiff or the defendant. For instance, if the plaintiff required the defendant to sign a non-disclosure agreement, this is a plaintiff factor since it indicates that the plaintiff was taking due measures to protect intellectual property. If the defendant learned of the intellectual property in a public forum, this is a defendant factor since it indicates that the defendant did not misappropriate the plaintiff's property. We only discuss the base level factors for these are most closely associated with the linguistic factor indicators of the text, and the intermediate and higher level factors are inferred from the base level factors. Of the 27, we have investigated the following six factors:[3]

Pro Plaintiff Factors

- F6 Plaintiff-adopted-security-measures
- F7 Defendant-hired-plaintiff-employee
- F21 Defendant-knew-information-confidential

Pro Defendant Factors

- F1 Plaintiff-disclosed-information-in-negotiations
- F10 Plaintiff-disclosed-information-to-outsiders
- F27 Plaintiff-disclosed-information-in-public-forum

The rationale for the selection of cases and factors is as follows. We only have a fragmentary list of the factors which appear in the cases available to us ((Aleven, 1997) and (Chorley, 2007)). We want to find at least one plaintiff factor and one defendant factor in each case, with some factors appearing in more than one case, though we have not done an analysis with respect to every factor in this set of cases. In particular, we find the following, which also indicates the winning side, where we only include those factors under investigation:

- FMC Outcome: Plaintiff
  - Pro Plaintiff: F6, F7
  - Pro Defendant: F10
- Goldberg Outcome: Plaintiff
  - Pro Plaintiff: F21
  - Pro Defendant: F1, F10, F27
- Midland Outcome: Plaintiff
  - Pro Plaintiff: F7
  - Pro Defendant: F10, F27
- Trandes Outcome: Plaintiff
  - Pro Plaintiff: F4, F6
  - Pro Defendant: F1, F10

The objective of the automated and manual annotation tasks is to automatically or manually identify material in the text of the case which is associated with the factor. We then compare and contrast the results. The manually annotated cases, suitably refined and expanded, provides a *gold standard* against which to evaluate the automated techniques. The development of both annotation approaches allows us to iteratively develop the overall objective of a well-developed factor analysis for this set of legal cases.

3. Methodology

In this section, we outline our methodology for developing the annotations, then report and discuss the results.

3.1. GATE

The techniques described in this paper rely on the GATE architecture (Cunningham et al., 2002). GATE is a framework for language engineering applications, which supports efficient and robust text processing. Overall, the GATE platform consists of two main functionalities:

- GATE Developer is an open source desktop application written in JAVA that provides a user interface for professional linguists and text engineers to bring together a wide variety of text analysis tools and apply them to a document or set of documents. GATE Developer incorporates many NLP tools as plug-ins. Some have been developed in-house, others have been written specifically for GATE and others have been ported from stand-alone open-source tools.
- GATE TeamWare is a web-based management platform for collaborative annotation and curation. It delivers a multi-function user interface over the internet for viewing, adding and editing text annotations. It allows the specification, managing and monitoring of the workflow of the collaborative text annotation work over the internet, and structures the contributions from different actors (human and machine) into clearly-defined roles.

For our purposes, we have applied the following modules in order to our texts, each module providing input to the next; the last two modules are explained further below:

[3] We have maintained the numbering of the factors, but changed the labels in order to make them more informative for the manual annotation task.

- Sentence splitter, which splits the text into sentences.
- Tokeniser, which identifies basic 'tokens' or words in the text.
- Part of speech tagger, which associates tokens with parts of speech such as noun, verb, and adjective.
- Morphological analyser, which lemmatises the tokens to provide words in their *root* form. This allows us to work with uniform word forms rather than taking into consideration morphological variants as in *sing*, *sung*, and *sang*.
- Gazetteer, which is a list of lists, where each list is comprised of words that are associated with a central concept.
- Java Annotation Patterns Engine (JAPE), which enable rules to be written with annotations and regular expressions as input, and annotations as output.

In the next sections, we detail first the construction of the gazetteer lists and JAPE rules followed by results. Then we present how we worked with GATE TeamWare and our results. In Figure 2, we represent the workflow .

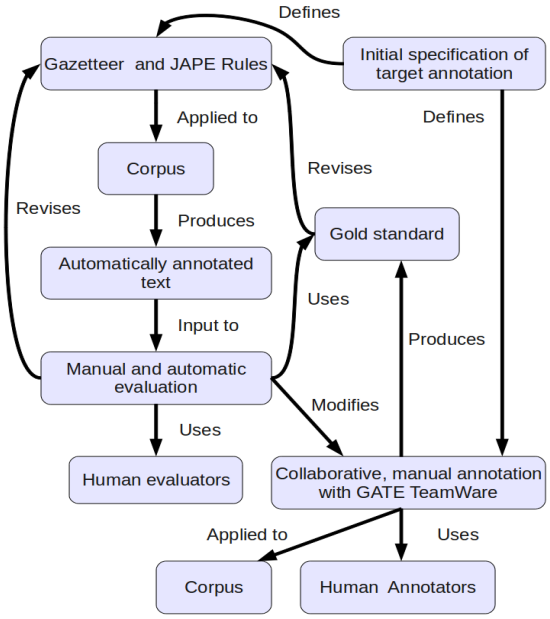

Figure 2: A Workflow Diagram

3.2. Development of GATE elements

Given our materials, the method we employed is knowledge heavy, bottom up, and cascading; we focus on the development of gazetteer lists and JAPE rules. It is knowledge heavy in the sense that in making the lists and rules, we have taken salient concepts from descriptions of case factors, used information about word relationships, and provided for alternative orders of terms. We have taken the descriptions of the case factors in (Aleven, 1997), identified key concepts that relate to the factor, used WordNet to identify semantically related terms, then used that list of terms to define rules that provide for bottom level concept annotations. These bottom level annotations are then be used to define rules for compound annotations. We use tools in GATE to view or extract the occurrences of the annotations. As pointed out earlier, the annotations are taken to be indicative of the factors to a lesser or greater extent, where the factors are events or topics which are linguistically expressed. By iteratively refining the lists and rules, our automated processing will, we expect, approximate manual identification of the precise linguistic realisations.

3.3. A sample factor description

A sample factor presentation from (Aleven, 1997, p. 242) follows. As discussed, we are only considering the base level factors. The factor presentation contains the index (F1), a label `Disclosure-In-Negotiations`, the side favoured if the factor holds (d represents defendent and p plaintiff), a description comprised of the event or situation along with some explanatory meaning relevant to the case, and some indications of when the factor does and does not apply (which are not always given for every factor).

- F1 Disclosure-In-Negotiations (d)
- Description: Plaintiff disclosed its product information in negotiations with defendant. This factor shows that defendant apparently obtained information by fair means. Also, it shows that plaintiff showed a lack of interest in maintaining the secrecy of its information.
- The factor applies if: Plaintiff disclosed the information to defendant in the context of negotiating a joint venture, licensing agreement, sale of a business, etc.
- The factor does not apply if: Defendant acquired knowledge of plaintiffs information in the course of employment by plaintiff.

Other factor presentations are similar.

3.4. Manual term extraction

From the factor presentation, we have manually extracted the most salient terms and simplified the presentation (primarily for use in the manual annotation task). For instance, we extract the following lemmatised terms and phrases from F1:

> plaintiff, disclose, product, information, negotiation, defendant, obtain, fair means, show, lack of interest, maintain, secrecy, joint venture, licensing agreement, sale of a business, acquire, knowledge, employment

We simplified the presentation of the factor:

- F1 Plaintiff-disclosed-information-in-negotiations
- Plaintiff disclosed information during negotiations with defendant. The defendant fairly obtained the information and the plaintiff was not interested to maintain the information as a secret.

- Applies if the plaintiff disclosed the information to defendant during negotiations for a joint venture, licensing agreement, sale of a business, etc..
- Does not apply if the defendant learned the information while employed by plaintiff.

3.5. Expansion of terms to create gazetteer

From the extracted terms and phrases of the factor presentations, we made classes of synonymous terms. Then we consulted WordNet to identify synonymous or salient terms that relate to some legally applicable concept. For example, for "disclosure", we found the following:

> announce, betray, break, bring out, communicate, confide, disclose, discover, divulge, expose, give away, impart, inform, leak, let on, let out, make known, pass on, reveal, tell, announcement, betrayal, communication, confidence, disclosure, divulgance, exposure

These terms comprise the strings in a gazetteer list, `disclosure.lst` with majorType `disclosure`. This means that during the lookup phase of processing, the gazetteer lists are consulted, and terms (i.e. Tokens) which appear on a list are annotated with Lookup as the majorType from the relevant list; that is, when GATE finds a token such as "confide" in the text, GATE annotates the token with Lookup = `disclose`. Thus, the function of the gazetteer lists is to provide a *cover concept* for related terms that can be used by subsequent annotation processes. As an initial development, this manual method can be used to *seed* automated methods to identify further relevant terms; alternatively, other resources can be drawn on to elaborate or refine the underlying lists. However, we do not presume prescriptive automation, where the content of the lists is fixed by the authors; rather, the lists will be refined and elaborated in a process of community development. In addition, list development may be related to ontological development, where the major type serves as a concept cover term for terms that may vary in their lexical semantics.

The context dependent interpretation of lexical items is a significant problem. In legal cases, we have terms that have a functional role. For example, whether an object is a weapon or "just" an object (such as a pen) depends on the context and the actions; similarly, individuals and organisations have a functional role, an individual may be a plaintiff in one case and a defendant in another. This is often a problem in dealing with natural objects versus socially defined objects as well as when we are dealing with fixed versus flexible reference. For our purposes, we put these significant issues aside in order to begin to develop the means to identify factors; our view is that we can better address functional roles where we manually provide annotated information and reason with the information in an ontology.

Finally, there are issues of polysemy of terms. However, we are addressing a highly restrictive domain (reports of decisions in case law) rather than an entirely open domain. Moreover, in a legal context, it is crucial to disambiguate terms and keep the interpretation of terms fixed. Thus, we believe these issues, while important to attend to where they occur, are not salient in our domain.

3.6. The bottom level annotation from a JAPE rule

Once Tokens have a Lookup value, we create JAPE rules for each Lookup value, which creates annotations that appear in GATE's annotation set. For instance, given the Lookup majorType `disclosure`, we create an annotation `Disclosure`.

```
Rule:  DisclosureFactor01
({Lookup.majorType == ``disclosure''}
):temp
-->
:temp.Disclosure = {rule =
``DisclosureFactor01''}
```

The annotations are the building blocks of a language for compound JAPE rules which annotate phrases or sentences with respect to two or more basic annotations.

3.7. Compound rules

In the following, we have an example compound JAPE rule annotates sequences of tokens with the `Disclosure` annotation, followed by zero or more `Tokens` within a sentence (given by ({Token, !Split})*), followed by the Information annotation. The whole text span is annotated `DisclosureInformationXY`.

```
Rule:  DisclosureInformationXY
({Disclosure}
({Token, !Split})*
{Information}
):temp
-->
:temp.DisclosureInformationXY = {rule =
``DisclosureInformationXY''}
```

The linear order of the annotations is crucial: in the rule above, we can only find `Disclosure` followed by `Information`; this can appear where we have, for example, an active sentence such as *Bill disclosed the information*. However, were we to have some alternative order of the annotations, then the rule above would not succeed. Therefore, we must write another rule to take into account the alternative order, where XY as above indicates one order, YX indicates another order. For every pair of annotations we want to annotate as a compound, we need at least two rules; for a rule containing 3 elements, we might require 6 rules for all the alternative orders; however, in practice, this is not clearly required. Once we have all the alternative orders, we write a "cover" rule, which makes the order irrelevant as in:

```
Rule:  DisclosureInformation
({DisclosureInformationXY} |
{DisclosureInformationYX}
):temp
-->
:temp.DisclosureInformation = {rule =
``DisclosureInformation''}
```

3.8. Factor rules

Using either bottom level or compound level annotations, we define the highest level annotation rule which is intended to annotate a factor. For example, for the target factor `F1 Plaintiff-disclosed-information-in-negotiations`, along with annotations for `Dislosure`, `Information`, and `Negotiation`, we provide a rule for one order of the bottom level annotations:

```
Rule: DisclosureInformation-
NegotiationXYZ
({Disclosure}
({Token, !Split})*
{Information}
({Token, !Split})*
{Negotiation}
):temp
-->
:temp.DisclosureInformationNegotiation-
XYZ = {rule = ''DisclosureInformation-
NegotiationXYZ''}
```

We create rules for all relevant alternative orders and provide a "cover" rule such as the following for `DisclosureInformationNegotiation`, which for Factor F1:

```
Rule: DisclosureInformationDisseminate
( {DisclosureInformationDisseminate-
TempZYX} |
{DisclosureInformationDisseminate-
TempZXY} |
{DisclosureInformationDisseminate-
TempYXZ} |
{DisclosureInformationDisseminate-
TempYZX} |
{DisclosureInformationDisseminate-
TempXYZ} |
{DisclosureInformationDisseminate-
TempXZY}
):temp
-->
:temp.DisclosureInformationDisseminate
= {rule = ''DisclosureInformation-
Disseminate''}
```

3.9. Additional gazetteer lists and factor rules

In addition, we have 37 gazetteer lists along with their related JAPE rules. We have the list name, sample elements, and the annotation.

- usehave.lst: have, use, adopt: UseHave
- confidential.lst: confidential: Confidential
- disclosure.lst: disclosure: Disclosure
- disseminate.lst: disseminate: Disseminate
- form-employee.lst: formemployee: FormEmployee
- hire.lst: hire: Hire
- information.lst: information: Information
- know.lst: know: Know
- negotiate.lst: negotiate: Negotiate
- outsider.lst: outsider: Outsider
- secureinfo.lst: secureinfo: SecureInformation

We have factor rules constructed from this language and homogenising over word order:

- DisclosureInformationNegotiation:
  F1 Plaintiff-disclosed-information-in-negotiations
- DisclosureInformationOutsider:
  F10 Plaintiff-disclosed-information-to-outsider
- DisclosureInformationDisseminate:
  F27 Plaintiff-disclosed-information-in-public-forum
- UseHaveSecureInformation:
  F6 Plaintiff-adopted-security-measures
- HireFormEmployee:
  F7 Defendant-hired-plaintiff-employee
- KnowConfidentialInformation:
  F21 Defendant-knew-information-confidential

In the rules, we have not identified the entities for plaintiff or defendant, which are functional roles in a case where the role an entity plays may vary from case to case. In general, as this aspect of cases is a complex problem in itself, we have focused on the identification of key information about the factors in order to highlight candidate spans of texts.

3.10. Results

In this section, we report the results of running our gazetteer lists and JAPE rules over our corpus. The results are given as output using the GATE Annotations-in-Context tool (ANNIC). ANNIC allows one to index and search a corpus by annotation: ANNIC produces the textual span covered by the annotation, the textual spans on either side of the annotated span, the source document for each span, and the number of occurences of the annotation in the corpus. In addition, one can search for bottom level annotations as well as combinations of them to create complex queries. We provide the results from bottom level annotations to compound factor annotations.
In Table 1, we present results for bottom level annotations, indicating the numbers of occurrences per case. Recall that these annotations are given by rule from the gazetteer lookup of lists. In turn, the gazetteer lists are intended to represent concepts given by a range of lexical items. Thus, the results are to be interpreted as the linguistic indication of the concept in the case.[4]

[4] *Secure information* is given in a list by phrases such as *invention agreement*, though these could have been constructed by rule.

Bottom Level	FMC	Gold.	Mid.	Tra.
Confidential	0	18	2	4
Disclosure	3	61	5	21
Disseminate	4	49	3	3
FormEmployee	2	3	5	11
Hire	0	4	7	1
Information	9	91	23	125
Know	0	7	1	17
Negotiate	10	4	4	
Outsider	6	4	5	3
SecureInformation	5	0	0	0
UseHave	37	109		72

Table 1: Bottom level annotations in cases

There are clearly relationships between these terms which merit further independent elaboration. For example, `Disseminate`, `Negotiate`, and `Outsider` relate to communications in which the parties with the information make it available to a wider audience. The properties and contexts which differentiate them will be left for future discussion.

In Table 2, we present results for pairs of annotations which are relevant to factors of three base annotations. We have used ANNIC to create searches for select pairs of annotations with 15 tokens between them without intervening sentence splitters; the results sum both orders of the pair. The results are to be interpreted as the linguistic indication of the pair of concepts in a sentence in the case.

Bottom Level	FMC	Gold.	Mid.	Tra.
Conf., Info.	0	19	1	0
Discl., Info.	0	37	7	6
Diss., Info.	2	16	3	2
Info., Out.	1	2	0	0

Table 2: Pairwise annotations in cases

The results are an overestimation in that we have not removed overlapping annotations; that is, for example, for `Information` and `Outsider`, we find both text spans "information regarding prospective customers" and "fact make information regarding prospective customers" in one case, where the latter contains the former. ANNIC does not identify the minimal span.

In Table 3, we give the factor (using the factor index) and the number of occurrences of the annotation with respect to cases in which those occurrences appear. Recall that the factors are compounds of two or more bottom level terms.

Other than for F27, disclosure of information in a public forum, the results for factors are poor, though the trend is clear – the more combinations of bottom level annotations, the fewer compound annotations. In a sense, the results are surprising. Given that we have taken cases which are reported to contain the relevant factors, that we have used and broadened the terms of the factor descriptions, and that we have overlooked a range of issues that might interfere with the results, one might have expected an *overgeneration* of results.

Factor	FMC	Gold.	Mid.	Tra.
F1	0	0	0	1
F6	1	0	0	0
F7	0	0	1	0
F10	0	0	0	0
F21	0	0	0	0
F27	0	20	5	0

Table 3: Factors in cases

To be clear, consider a range of potentially interfering issues. First, we have not taken into account reports of a fact pattern, which indicates that it holds in a case, from discussions about the concept, which do not imply the fact holds. For example, fact patterns appear as subordinate clauses under the scope of propositional attitudes or speech acts such as *believe*, *allege*, *claim*. Similarly, we have not considered fact patterns under the scope of negation *not* or terms with negative implication such as *fail* or *deny*, which again do not imply that the fact pattern holds. We have not filtered results with respect to syntactic structure. Nor have we constrained the results with respect to parties in a case. Finally, recall that our results abstract over word order.

There are a variety of ways that the results could be incrementally improved. First, we could augment the terms in the gazetteer lists given evidence from the language of the corpus and relative to the manual annotation task. In this regard, it is essential to build an accurate, richly annotated corpus manually. We should also examine the role of anaphora, ellipsis, syntactic phrases, and terms distributed across sentence boundaries. Finally, the results (in combination with the manual annotation) raise questions about the initial factor ascriptions given in (Aleven, 1997) and (Aleven, 2003); our approach does not verify the expected annotation results. One explanation is that the four cases under examination are cases on appeal rather than on first instance, which means that the facts of the cases are not under discussion but rather a point of law or interpretation. Thus, the cases in our corpus are fact pattern poor, and it remains to be verified whether the facts attributed to the cases hold or have rather been imported from the cases of first instance (e.g. by some referential mechanism). Even were this so, there is the substantive issue of whether it is correct to import such factors since the decision about the case on appeal may not rest primarily on reasoning about the factors.

At bottom, our approach emphasises an interesting question that is not highlighted in machine learning or statistical approaches – what do judges, lawyers, jurors, and law students know when they know to identify a factor in the text? Clearly, they rely on overt linguistic indicators, structure, and semantic interpretations which interact with domain knowledge. The question is important to address, for unlike other applications of information extraction where a "black box" result may be acceptable, it is highly relevant in the legal domain to provide an explicit justification and explication for a legally binding decision and to represent this in the text of the decision for subsequent reuse in case based

reasoning. Thus, despite the current results, there is strong reason to continue to investigate the phenomena. The question also touches on the representation of legal knowledge. In our annotation experiment, we limit ourselves to explicit lexical realisations of factors. However, legal knowledge would also be represented with ontologies and rules, providing intermediate level, non-lexicalized, implicit knowledge necessary for succesful semantic factor annotation.

4. Manual case factor annotation

The manual annotation task performed in GATE TeamWare serves several purposes. First, it creates a gold standard, on the basis of which the predictive power of the automatically annotated low-level (e.g. bottom level) factors for high-level factor annotation can be evaluated. Second, we cannot a priori expect a full correspondence between the low-level and high-level factors. Therefore, we should also regard the manual annotation as an exploration of the interaction between the various levels of factor annotation. Thirdly, the quality of the annotations and the inter-annotator agreement can give an indication of the quality of the factors themselves. Given the rather incomplete, vaguely defined, and overlapping nature of the factors that we have available, we may expect lack of clarity amongst the annotators.

Figure 3: GATE TeamWare with low and high level factor annotations

The annotation task itself is rather complex as well. The annotators must be familiar with the semantics of the factors, and ideally agree on the exact text spans for each factor annotation. Because factors are expressed in flexible and non-predictable ways it cannot be expected that annotators agree to a high level on the exact boundaries of the linguistic text element expressing the factors under examination. In fact, this is borne out by the inter-annotator agreement results. TeamWare enables the computation of inter-annotator agreement in several ways. We have chosen precision, recall and F1 measure. Three documents yield zero values for all, whereas one document gives 0.5 scores for precision, recall and F1. In conclusion, we see little or no agreement between the annotators of the high-level factors. In many cases, the annotators' results are complementary rather than overlapping. We consider this an indication of the difficulty of spotting the exact lexicalisations of the complex concepts expressed by the factors. As Figure 3 shows, the low inter-annotator agreement is also partly due to overlapping, but non-identical text spans. Annotator one chose a larger text span, which includes the text span selected by Annotator two. In our opinion, this is again indicative of the highly non-trivial nature of the task.

4.1. Comparison of high-level and low-level factor annotation

In this section, we evaluate the correspondence between the factors of different levels in order to judge the predictive strength of low level factors (Low) for the selection of high level factors (High). High level factors tend to share low and compound level factors within their annotation spans. Standardly, evaluation mechanisms of precision and recall presuppose a dependency between low and high level factors which is binary – indicative or not indicative. However, in our bottom-up approach, we cannot assume that "indicativeness" is a binary notion. Rather, we postulate levels of indicativeness, where the frequency of text span enclosure is the observable measure.
Of the low level factors listed in Table 1 and the compound level factors described in Sections 3.8. and 3.9., the factors that uniquely indicate high level factors in our small corpus are in Table 4, given frequency of occurrence of the low level factor in the high level factor (Freq.) and percentage of occurrence in the corpus (Perc.).

Low	High	Freq.	Perc.
UseHaveSecureInformation	F6	1	100
SecureInformation	F6	1	20
Fair	F6	2	8.6
Appellee	F7	1	10
Defendant	F7	1	2
Hire	F7	2	16.6
Agreement	F7	2	50
Plaintiff	F10	2	3.7
Know	F21	2	8
Confidential	F21	2	4
ConfidentialInformation-TempXY	F21	1	6.6
ConfidentialInformation	F21	1	5.3

Table 4: Unique lower level indicators of high level factors

The ones with a high percentage of occurrence in the corpus, such as the compound level factor `UseHaveSecureInformation` and the low level factor `Agreement`, can be seen as strong indicators of the high level factors F6 and F7 respectively. It needs to be stressed that given the size of the corpus, we cannot make strong claims about how representative the results are of the

sub-domain we are targeting. However, a qualitative evaluation suggests that `UseHaveSecureInformation` and `SecureInformation` are both more tightly related to F6 than any other high level factor. The non-unique low level factors are shared amongst the high level factors for several reasons:

- Their presence is not indicative of the semantics of the high level factor.
- They express meaning components that are shared by the high level factors, and therefore point to vague distinctions between these high level factors. If the latter applies, an incremental refinement of the factors is needed.

5. Related work

(Brüninghaus and Ashley, 2003), (Brüninghaus and Ashley, 2005), and (Ashley and Brüninghaus, 2009) consider both knowledge representation and reasoning along with textual information processing of case factors. A system is proposed to classify cases with respect to the facts and then to predict the outcome of a case. We consider the text analysis.

(Ashley and Brüninghaus, 2009) apply NLP techniques to a squib rather than the original text of the case decision; a squib is a manually constructed summary of the case which represents the factors of the case along with factor indices; the factors are text fragments that are incorporated into a narrative. From a set of squibs, a list of positive and negative statements of each factor is manually constructed (the learning set); from the list, machine learning techniques are applied to "acquire" a classifier (a pattern) for each factor. The classifiers are applied to the test set of squibs, and each text is classified as to the factors contained within it. A *nearest-neighbour* machine learning algorithm is applied to a learning set of squibs, where the classifying pattern is compared to sentences in the test set to find sentences most similar to the classifying pattern. The success of the classification is measured against a gold standard of squibs, which have been manually classified.

Given that the results rely on the classifying pattern, several alternative representations for the learning set are considered. This means that prior to applying the machine learning algorithm, the squibs are further preprocessed using a range of NLP techniques. The three representations are:

- bag of words - the degree to which one squib is similar to another squib in terms of the lexical items in each.
- replacement - the name of an individual is replaced by their functional role in the case, e.g. IBM for plaintiff.
- "propositional patterns" - 4 pair-wise part of speech patterns such as 'subject-verb', 'verb-object', 'verb-prepositional phrase' and 'verb-adjective'. A thesaurus creates alternative patterns using synonymous words within a pattern.

(Ashley and Brüninghaus, 2009) report that the F-measure, which measures the accuracy and completeness of the coverage (1 is perfect accuracy and completeness), of any of the classification tasks is very low, for one experiment it was below 0.3. However, the reports are predominately given indirectly in terms of the impact of the representations on the results of Issue-based Prediction (IBP) of case decisions, which is a case based reasoning system.

Our approach differs from (Ashley and Brüninghaus, 2009) in several respects. First, we work with original, unstructured text rather than structured text which does not address the knowledge bottleneck at the point of identifying the factors from unstructured text. However, using structured text does have obvious advantages to unstructured text, but only to the extent that results can be extended to cover unstructured text. Second, we work with the conceptual components of the case factors (bottom level and compound annotations), and we do not apply parsing and entity extraction (e.g. party names and product information), which do not clearly provide advantages to factor identification. Indeed, since our approach generalises over the bag of words approach (incorporating a thesaurus directly to form concepts), and neither roles nor syntactic relations are relevant to further restrict output, we might have expected overgeneralised results, as we discussed earlier. Third, (Ashley and Brüninghaus, 2009) classify case squibs with respect to factors, and the factors themselves are not annotated, which implies that one cannot extract the factors per se. In our approach, factors are explicitly annotated and can be extracted. As we do not have a well-defined gold standard, we do not apply machine learning techniques, which would be premature. The source cases, squibs, and gold standard of (Ashley and Brüninghaus, 2009) are not available for public evaluation, so it is difficult to independently verify the results or contribute to the development of a factor extraction system. In contrast, we work with tools and material that promote a community development process to refine the the gazetteers and JAPE rules as well as to develop a consensus gold standard. Finally, a machine learning approach provides classifiers which may be opaque to users and which need not represent the knowledge that law students or legal professionals bring to the task of factor identification. In our approach of bottom level and compound concepts, important aspects of legal knowledge are made explicit.

6. Discussion and conclusion

We have outlined and reported an approach to annotation of legal case factors in full text decisions. It is bottom-up, starting with concepts over a range of lexical items, then constructing more complex factors from the concepts. While the results of this initial study are poor, they highlight a range of issues that can be addressed in further research – augmenting the gazetteer lists, constraining contexts under the scope of negation or propositional attitudes and speech acts, taking into account the role of ellipsis and anaphora, as well as the difference between cases of first instance and cases on appeal. We have also conducted an online, collaborative annotation task. The results indicate that further refinement of the task is required.

However, overall, we have defined a clear, well-defined, open workflow for building an annotation and extraction system for legal case factors which supports iterative refine-

ment through a collaborative process. Didactically speaking, it would be of great interest to involve law school students and legal professionals in the task of building the lists, JAPE rules, and carrying out online annotation tasks, for not only would this refine the tool, but it would encourage participants to focus on close textual analysis of the cases, which is a core capacity of every lawyer.

Given a gold standard of texts and a method to add annotated cases to the case base, we could use the extracted factors as input to a case based reasoning system such as IBP or the argument schemes of (Wyner and Bench-Capon, 2007). In addition, cases with annotated factors (in an XML compatible format) could be used for Semantic Web applications such as information extraction, querying, and reasoning with cases over the internet.

The scale of the experiment is small in terms of number of documents and of annotators. It is clear that this is just a feasibility study. A real gold standard should be created by a larger number of annotators, which will also yield statistically more reliable correspondences between lower level and higher level factors.

In future work, we will apply machine learning techniques, term extraction, ontology construction, as well as experiment with the role of syntactic structure to improve results.

7. Acknowledgements

During the writing of this paper, the first author was in part supported by the IMPACT Project (Integrated Method for Policy making using Argument modelling and Computer assisted Text analysis) FP7 Grant Agreement No. 247228.

8. References

Vincent Aleven. 1997. *Teaching case-based argumentation through a model and examples*. Ph.D. thesis, University of Pittsburgh.

Vincent Aleven. 2003. Using background knowledge in case-based legal reasoning: A computational model and an intelligent learning environment. *Artificial Intelligence*, 150:183–237.

Kevin D. Ashley and Stefanie Brüninghaus. 2009. Automatically classifying case texts and predicting outcomes. *Artif. Intell. Law*, 17(2):125–165.

Kevin Ashley. 1990. *Modelling Legal Argument: Reasoning with Cases and Hypotheticals*. Bradford Books/MIT Press, Cambridge, MA.

Stefanie Brüninghaus and Kevin D. Ashley. 2003. Predicting the outcome of case-based legal arguments. In Giovanni Sartor, editor, *ICAIL'03: Proceedings of the 9th International Conference on Artificial Intelligence and Law*, pages 233–242, Edinburgh, United Kingdom. ACM Press: New York, NY.

Stefanie Brüninghaus and Kevin D. Ashley. 2005. Generating legal arguments and predictions from case texts. In *ICAIL 2005*, pages 65–74, New York, NY, USA. ACM Press.

Alison Chorley. 2007. *Reasoning with Legal Cases seen as Theory Construction*. Ph.D. thesis, University of Liverpool, Department of Computer Science, Liverpool, UK.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. 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)*.

Diana E. Forsythe and Bruce G. Buchanan. 1993. Knowledge acquisition for expert systems: some pitfalls and suggestions. In *Readings in knowledge acquisition and learning: automating the construction and improvement of expert systems*, pages 117–124. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Ben Hachey and Claire Grover. 2006. Extractive summarisation of legal texts. *Artificial Intelligence and Law*, 14(4):305–345.

Carole Hafner. 1987. Conceptual organization of case law knowledge bases. In *ICAIL '87: Proceedings of the 1st International Conference on Artificial Intelligence and Law*, pages 35–42, New York, NY, USA. ACM.

Peter Jackson, Khalid Al-Kofahi, Alex Tyrell, and Arun Vachher. 2003. Information extraction from case law and retrieval of prior cases. *Artificial Intelligence*, 150(1-2):239–290, November.

Guiraude Lame. 2004. Using nlp techniques to identify legal ontology components: Concepts and relations. *Artificial Intelligence and Law*, 12(4):379–396.

K. Tamsin Maxwell, Jon Oberlander, and Victor Lavrenko. 2009. Evaluation of semantic events for legal case retrieval. In *ESAIR '09: Proceedings of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 39–41, New York, NY, USA. ACM.

Diana Maynard, Yaoyong Li, and Wim Peters. 2008. NLP techniques for term extraction and ontology population. In *Proceeding of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 107–127, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Marie-Francine Moens, Caroline Uyttendaele, and Jos Dumortier. 1997. Abstracting of legal cases: the salomon experience. In *ICAIL '97: Proceedings of the 6th International Conference on Artificial Intelligence and Law*, pages 114–122, New York, NY, USA. ACM.

Wim Peters. 2009. Text-based legal ontology enrichment. In *Proceedings of the workshop on Legal Ontologies and AI Techniques*, Barcelona, Spain.

Edwina L. Rissland, David B. Skalak, and M. Timur Friedman. 1996. BankXX: Supporting legal arguments through heuristic retrieval. *Artificial Intelligence and Law*, 4(1):1–71.

Edwina L. Rissland, Kevin D. Ashley, and L. Karl Branting. 2006. Case-based reasoning and law. *The Knowledge Engineering Review*, 20:293–298.

Adam Wyner and Trevor Bench-Capon. 2007. Argument schemes for legal case-based reasoning. In Arno R. Lodder and Laurens Mommers, editors, *Legal Knowledge and Information Systems. JURIX 2007*, pages 139–149, Amsterdam. IOS Press.

Adam Wyner. 2008. An ontology in OWL for legal case-based reasoning. *Artificial Intelligence and Law*, 16(4):361–387, December.

Legal Rules Learning based on a Semantic Model for Legislation

Enrico Francesconi

ITTIG-CNR, via de' Barucci 20, Florence (Italy)
francesconi@ittig.cnr.it

Abstract

Legal rules extraction from legislative texts can be an effective method to make it easier the implementation of rules-based systems for legal assessment and reasoning, as well as for implementing advanced search and retrieval systems for legislative documents. In this paper machine learning and NLP techniques are used for extracting legal rules on the basis of a semantic model for legislative texts, which is oriented to knowledge reusability and sharing. Moreover the identified entities of the regulated domain can be a starting point to a bottom-up implementation of domain ontologies. This approach is aimed at giving a contribution to bridge the gap between consensus and authoritativeness in legal knowledge representation.

1. Introduction

Knowledge modelling represents a structural pre-condition for implementing the Semantic Web concept as well as intelligent systems dealing with legal information (Breuker et al., 2009). In literature different approaches are proposed for knowledge resources implementation: they can be ranged from top-down, bottom-up, as well as combined middle-out approaches having complementary characteristics. As (Uschold and Grüninger, 1996) pointed out, bottom-up approaches tend to result in a very high level of detail, while efforts and re-work on the selected concepts tend to be increased, as well as it is difficult to spot commonality between them. On the other hand top-down approaches allow better control of details, but can provide arbitrary high-level categories (Uschold and Grüninger, 1996).

In the legal domain any chosen approach (bottom-up, top-down or middle-out) to knowledge resource implementation present a further trade-off between *consensus* and *authoritativeness* to be considered. Consensus is an issue faced in knowledge representation in general (Gangemi et al., 2002), since ontological conceptualization has to be shared between stakeholders (Studer et al., 1998). Different approaches have been undertaken to reach consensus in legal knowledge representation: for example the *common-sense terms* approach (Hoekstra et al., 2009), based on common sense understanding of the terminology identifying concepts, as well as the *folksonomy* approach[1] based on social and collaborative activities of concepts selection and categorization (Gruber, 2006).

Knowledge representation in the legal domain, however, shows peculiarities due to the importance of authoritative systems based on legal rules for legal assessment and reasoning (Breuker et al., 2008), or advanced search engines able to retrieve not just documents but also the contained norms (Biagioli and Turchi, 2005). Both common-sense terms and folksonomy approaches are well suited to reach *consensus* on domain concepts, however, when applied to the description of *legal rules*, the gap between consensus and authoritativeness is usually emphasized. For example, by the common-sense terms approach, social and communicative words typical of the legal domain can be provided (Breuker and Hoekstra, 2004a): experts may provide rules description on entities and translate them into technical terminology (Hoekstra et al., 2009), but this activity might reduce *consensus*. Similarly, in the folksonomy approach stakeholders may provide description of rules regulating entities, which might reduce *authoritativeness*.

Nowadays a very active research area is represented by knowledge acquisition from texts (Buitelaar et al., 2005), since they still represent the most widely used communication medium on the Web. This approach can play an important role in legal knowledge acquisition, since written text is the most widely used way of communicating legal matters (Lame, 2005). Knowledge acquisition techniques can be used for implementing taxonomies or suggesting concepts for upper level ontologies, mainly hand-crafted by domain experts, and for identifying *legal rules* (Lame, 2005; Walter and Pinkal, 2009). In this paper an approach to support the acquisition of legal rules contained in legislative documents is presented: it is based on a semantic model for legislation and implemented by knowledge extraction techniques over legislative texts. This approach is targeted to provide a contribution to bridge the gap between consensus and authoritativeness in legal rules representation. Consensus in this context is not related to the acceptance of legal rules, which is not questionable, but to reaching the widest agreement on their semantic description. The proposed method addresses *consensus* by contributing to a uniform description of legal rules limiting human intervention; on the other hand *authoritativeness* is given by default, since rules are extracted from authoritative texts as the legislative ones.

This paper is organised as follows: in Section 2. an approach to legal rules modelling and acquisition is presented; in Section 3. a semantic model for legislative texts is introduced; in Section 4. a knowledge acquisition methodology is shown and tested; finally in Section 5. some conclusions on the benefits of the approach are reported.

2. An approach to legal rules modelling and acquisition

The proposed approach to legal knowledge acquisition is based on learning techniques targeted to extract *legal rules*

[1] Folksonomies (or social tagging mechanisms) have been widely implemented in knowledge sharing environments; the idea was first adopted by the social bookmarking site del.icio.us (2004) http://delicious.com

from text corpora. Legal rules are essentially "speech acts" (Searle, 1969) expressed in legislative texts regulating *entities* of a domain: their nature therefore justifies an approach aimed at the analysis of such texts.
Therefore, the proposed knowledge acquisition framework is based on a twofold approach:

1. Knowledge modelling: definition of a semantic model for legislative texts able to describe legal rules;
2. Knowledge acquisition: instantiation of legal rules, driven by the defined semantic model, through the analysis of legislative texts.

This approach traces a framework which combines top-down and bottom-up strategies: a top-down strategy provides a model for legal rules, while a bottom-up strategy identifies rules instances from legal texts. The bottom-up strategy in particular can be carried out manually or automatically. The manual bottom-up strategy consists, basically, in an analytic effort in which all the possible semantic distinctions among the textual components of a legislative text are identified. On the other hand the automatic (semi-automatic) bottom-up strategy consists in carrying out the previous activities being supported by tools able to classify Rules, according to the defined model, and to identify the Entities which Rules apply to. In this paper the automatic bottom-up strategy is presented.

3. Knowledge modelling

The proposed approach is based on knowledge modelling oriented to interoperability and reusability, and it is based on the separation betweeen types of knowledge to be represented by Semantic Web standards. The need of identifying and separating different types of knowledge has been widely addressed in literature (Casellas, 2008). For example (Breuker and Hoekstra, 2004b) criticised a common tendency to indiscriminately mix domain knowledge and knowledge on the process for which it is used, speaking of *epistemological promiscuity*. Similarly (Bylander and Chandrasekaran, 1987) and (Chandrasekaran, 1986) pointed out that usually knowledge representation is affected by the nature of the problem and by the applied inference strategy; this key-point is also referred by (Bylander and Chandrasekaran, 1987) as *interaction problem*: it is related to a discussion regarding whether knowledge about the domain and knowledge about reasoning on the domain should be represented independently. In this respect (Clancey, 1981) pointed out that the separation of both types of knowledge is a desirable feature, since it paves the way to knowledge sharing and reuse.
The knowledge model proposed in this work reflects these orientations and it is organized into the following two components:

1. Domain Independent Legal Knowledge (DILK)
2. Domain Knowledge (DK)

DILK is a semantic model of Rules expressed in legislative texts, while DK is any terminological or conceptual knowledge base (thesaurus, ontology, semantic network) able to provide information and relationships among the Entities of a regulated domain. The combination of DILK with one or more DKs is able to provide a formal characterization of Rules instances. For this reason we call the proposed methodology to legal knowledge modelling the *DILK-DK* approach.

3.1. DILK

DILK is conceived as a model for legal Rules, independently from the domain they apply to. In literature several models (classification) of legal rules have been proposed, from the traditional Hohfeldian theory of legal concepts (Hohfeld, 1978) until more recent legal philosophy theories due to Rawls (Rawls, 1955), Hart (Hart, 1961), Ross (Ross, 1968), Bentham (Bentham and Hart, 1970 1st ed 1872), Kelsen (Kelsen, 1991).
In this context a particular attention is worth to be given to the work of Biagioli (Biagioli, 1997). Combining the work of legal philosophers on rules classification with the Searlian theory of rules perceived as "speech acts", as well as the Raz's lesson (Raz, 1980) to perceive laws and regulations as a set of *provisions* carried by speech acts, Biagioli underlined two views or *profiles* according to which a legislative text can be perceived: a) a structural or *formal profile*, representing the traditional legislator habit of organizing legal texts in chapters, articles, paragraphs, etc.; b) a semantic or *functional profile*, considering legislative texts as composed by *provisions*, namely fragments of regulation (Biagioli, 1997) expressed by speech acts. Therefore a specific classification of legislative provisions was carried out by analysing legislative texts from a semantic point of view, and grouping provisions into two main families: *Rules* (introducing and defining entities or expressing deontic concepts) and *Amendments* (basically Rules on Rules). Rules are provisions which aim at regulating the reality considered by the including act. Adopting a typical law theory distinction, well expressed by Rawls, they consist in:

- *constitutive rules*: they introduce or assign a juridical profiles to entities of a regulated reality;
- *regulative rules*: they discipline actions ("rules on actions") or the substantial and procedural defaults ("remedies").

On the other hand, Amendments can be distinguished into:

- *content amendments*: they modify literally the content of a norm, or their meaning without literal changes (i.e. interpretation, extension, etc.);
- *temporal amendments*: they modify the times of a norm (come-into-force and efficacy time);
- *extension amendments*: they extend or reduce the cases on which the norm operates.

In Biagioli's model each provision type has specific arguments describing the roles of the entities which a provision type applies to (for example the *Bearer* is argument of a *Duty* provision). *Provision types* and related *Arguments* represent a semantic model for legislative texts (Biagioli,

1997). They can be considered as a sort of metadata scheme able to analytically describe fragments of legislative texts. For example, the following fragment of the Italian privacy law:

> *"A controller intending to process personal data falling within the scope of application of this act shall have to notify the "Garante" thereof, . . . "*

besides being considered as a part of the physical structure of a legislative text (a *paragraph*), can also be viewed as a component of the logical structure of it (a *provision*) and qualified as a *provision* of type *Duty*, whose arguments are:

Bearer:	"Controller";
Object:	"Process personal data"
Action:	"Notification"
Counterpart:	"Garante"

The specific textual anchorage of the Biagioli's model represents, in our point of view, its main strength. Since the DILK-DK approach aims at representing Rules instances as expressed in legislative texts, we consider the Biagioli's model, limited to the group of Rules, as a possible implementation of DILK. On the other hand "Rules on rules" affect indirectly the way how the reality is regulated, since they amend Rules in different respects (literally, temporarily, extensionally): therefore such provision types are not part of DILK model. On the other hand their effects on Rules instances has to be taken into account for knowledge acquisition purposes.

3.2. DK

In legislative texts *Entities* regulated by provisions are expressed by lexical units, however no additional information on such entities are provided. This information can be provided by a *Domain Knowledge* (DK) providing conceptualization of entities expressed by language-dependent lexical units[2]. Information on such entities at language-independent level, as well as their lexical manifestations in different languages have to be described by a DK. A possible architecture for describing a DK has been proposed within the DALOS project[3]; it is organized in two layers of abstraction:

- *Ontological layer*: conceptual modelling at language-independent level;
- *Lexical layer*: language-dependent lexical manifestations of the concepts at the Ontological layer.

More details on the DALOS DK architecture, as well as a possible implementation of it for the domain of consumer protection, can be found in (Agnoloni et al., 2009).

[2] "Typically regulations are not given in an empty environment; instead they make use of terminology and concepts which are relevant to the organisation and/or the aspect they seek to regulate. Thus, to be able to capture the meaning of regulations, one needs to encode not only the regulations themselves, but also the underlying ontological knowledge. This knowledge usually includes the terminology used, its basic structure, and integrity constraints that need to be satisfied." (Antoniou et al., 1999)

[3] http://www.dalosproject.eu

4. Knowledge acquisition

Knowledge acquisition within the DILK-DK framework consists of two main steps: 1) DILK instantiation, 2) DK construction.

4.1. DILK instantiation

The DILK instantiation phase is a bottom-up strategy for legislative text paragraphs classification into *provision types*, as well as specific lexical units identification, assigning them roles in terms of *provision arguments*. The automatic bottom-up strategy, here proposed, consists in using tools able to support the human activity of classifying provisions, as well as to extract their arguments. Three main steps can be foreseen:

- Collection of legislative texts and conversion into an XML format (Bacci et al., 2009)
- Automatic classification of legislative text paragraphs into provisions (Francesconi and Passerini, 2007)
- Automatic argument extraction (Biagioli et al., 2005)

Legislative documents are firstly collected and transformed into a jurisdiction-dependent XML standard (NormeInRete in Italy, Metalex in the Netherlands, etc.). For the Italian legislation a module called xmLegesMarker, of the xmLeges[4] software family, has been developed (Bacci et al., 2009): it is able to transform legacy contents into XML so to identify the formal structure of a legislative document.

4.1.1. Automatic classification of provisions

For the automatic classification of legislative text paragraphs into provison types, a tool called xmLegesClassifier of the xmLeges family has been developed. xmLegesClassifier has been implemented using a Multiclass Support Vector Machine (MSVM) approach, as the one reporting the best results in preliminary experiments with respect to other machine learning approaches (Francesconi and Passerini, 2007). With respect to (Francesconi and Passerini, 2007), in this work MSVM is tested on the Rules provision family, as first step of DILK instantiation. Documents are represented by vectors of weighted terms and some pre-processing operations are performed on pure words to increase their statistical qualities:

- Stemming on words in order to reduce them to their morphological root[5]
- Stopwords elimination
- Digits and non alphanumeric characters represented by a unique character (since they do not provide semantics to provision instances).

Moreover feature selection techniques are applied to reduce the number of terms to be considered, thus actually restricting the vocabulary to be employed (see e.g. (Yang and Pedersen, 1997)). We tried two simple methods:

[4] http://www.xmleges.org

[5] We employed the snowball software, available at http://www.snowball.tartarus.org/italian/stemmer

Class labels	Provision Types	Number of documents
c_0	Definition	10
c_1	Liability	39
c_2	Prohibition	13
c_3	Duty	59
c_4	Permission	15
c_5	Penalty	122

Table 1: **Dataset of provision types**

- An unsupervised *min frequency* threshold over the number of term occurences in the training set, so to eliminate terms with unrealiable statistics.
- A supervised threshold over the Information Gain (Quinlan, 1986) of terms, which measures how much a term discriminates between documents belonging to different classes. The Information Gain of term w is computed as:

$$ig(w) = H(D) - \frac{|D_w|}{|D|}H(D_w) - \frac{|D_{\bar{w}}|}{|D|}H(D_{\bar{w}})$$

where H is a function computing the entropy of a labelled set ($H(D) = \sum_{i=1}^{|C|} -p_i \; log_2(p_i)$, being p_i the portion of D belonging to provision type i), D_w is the set of training documents containing the term w, and $D_{\bar{w}}$ is the set of training documents not containing w. This method basically allows to select terms with the highest discriminatory power among a set of provision types.

Once basic terms have been defined, a vocabulary of terms $\mathcal{T}$ can be created from the set of training documents $\mathcal{D}$, containing all the terms which occur at least once in the set. A single document d is represented as a vector of weights $w_1, \ldots, w_{|\mathcal{T}|}$, where the weight w_i represents the amount of information which the i^{th} term of the vocabulary carries out with respect to the semantics of d. We tried different types of weights, with increasing degree of complexity:

- a *binary* weight $\delta(w, d)$: presence/absence of the term within a document;
- a *term-frequency* weight $tf(w, d)$: number of times a term occurs within the document (measure of its representativeness of a document content);
- a combination of *information gain* and *term-frequency* ($ig(w, d) * tf(w, d)$);
- a *tf-idf* (Buckley and Salton, 1988) weight: term specificity degree with respect to a document.

A wide range of experiments was conducted over a dataset made of 258 Rules instances, collected by legal experts, distributed among 6 provision classes (Tab. 1). After terms preprocessing, we tried a number of combinations of the document representation and feature selection strategies previously described. We employed a *leave-one-out* (loo) procedure for measuring performances of the different strategies and algorithms. For a dataset of n documents $D = \{d_1, \ldots, dn\}$, it consists of performing n runs of the learning algorithm, where for each run i the algorithm is trained on $D \setminus d_i$ and tested on the single left out document d_i. The loo accuracy is computed as the fraction of correct tests over the entire number of tests. Table 2 reports loo accuracy and train accuracy, which is computed as the average train accuracy over the loo runs, of the Multiclass Support Vector Machine algorithm for the different document representation and feature selection strategies. The first three columns (apart from the index one) represent possible preprocessing operations. The fourth column indicates the term weighting scheme employed (binary (δ), term frequency (tf), infogain * term frequency ($ig * tf$), term frequency-inverse document frequency (tf-idf)). The two following columns are for feature selection strategies: the unsupervised *min frequency* and the supervised *max infogain*, which actually indicates the number of terms to keep, after being ordered by Information Gain. Finally, the last two columns contain loo and train accuracies.

#	repl. digit	repl. alnum	use stem	weight scheme	min freq sel.	max IG sel.	loo acc (%)	train acc (%)
0	no	no	no	δ	2	500	89.53	100
1	yes	no	no	δ	2	500	88.76	100
2	yes	yes	no	δ	2	500	88.76	100
3	yes	yes	yes	tf	2	500	91.09	100
4	yes	yes	yes	tf-idf	2	500	89.15	100
5	yes	yes	yes	ig	2	500	89.15	100
6	yes	yes	yes	ig*tf	2	500	89.15	100
7	yes	yes	yes	δ	2	250	89.92	100
8	yes	yes	yes	δ	2	100	82.55	100
9	yes	yes	yes	δ	2	50	82.17	96.12
10	yes	yes	yes	δ	2	1000	90.31	100
11	yes	yes	yes	δ	0	500	92.24	100
12	yes	yes	yes	δ	2	500	92.64	100
13	yes	yes	yes	δ	5	500	92.24	100
14	yes	yes	yes	δ	10	500	89.92	100

Table 2: **Detailed results of MSVM algorithm for different document representation and feature selection strategies.**

While replacing digits or non alphanumeric characters does not improve performances, the use of stemming actually helps clustering terms with common semantics. The simpler binary weight scheme appears to work better than term frequency, probably for the small size, in terms of number of words, of the provisions in our training set; this fact makes statistics on the number of occurences of a term less reliable. Slight improvements can be obtained by performing feature selection with Information Gain, thus confirming how SVM algorithms are able to effectively handle quite large feature spaces.

Classes	c_0	c_1	c_2	c_3	c_4	c_5
c_0	122	0	0	0	0	0
c_1	1	9	4	0	1	0
c_2	0	3	55	0	1	0
c_3	2	0	1	6	1	0
c_4	1	1	3	0	8	0
c_5	0	0	0	0	0	39

Table 3: **Confusion matrix for the best MSVM classifier.**

Finally, Table 3 shows the confusion matrix for the best classifier, the MSVM indexed 12, reporting prediction details for individual classes. Rows indicate true classes, while columns indicate predicted ones. Note that most errors are obtained in classes with fewer documents, for which unrealiable statistics could be learned.

4.1.2. Automatic provision arguments extraction

A tool called xmLegesExtractor[6] (Biagioli et al., 2005) of the xmLeges family has been implemented for the automatic detection of provision arguments. xmLegesExtractor is realized as a suite of NLP tools for the automatic analysis of Italian texts (see (Bartolini et al., 2004)), specialized to cope with the specific stylistic conventions of the legal parlance. A first prototype takes in input legislative raw text paragraphs, coupled with the categorization provided by the xmLegesClassifier, and identifies text fragments (lexical units) corresponding to specific semantic roles, relevant for the different types of provisions (Fig. 1). The approach

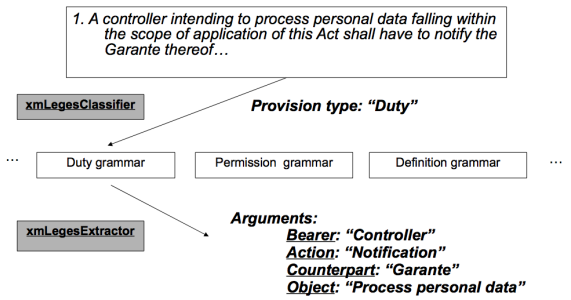

Figure 1: **xmLegesClassifier combined with the grammar approach used by xmLegesExtractor.**

follows a two–stage strategy. The first stage consists in a syntactic pre–processing which takes in input a text paragraph, which is tokenized and normalized for dates, abbreviations and multi–word expressions; the normalized text is then morphologically analyzed and lemmatized, using an Italian lexicon specialized for the analysis of legal language; finally, the text is POS-tagged and shallow parsed into non–recursive constituents called "chunks". The second stage consists in the identification of all the lexical units acting as arguments relevant to a specific provision type. It takes in input a chunked representation of legal text paragraphs, locating relevant patterns of chunks which represent entities with specific semantic roles within a provision type instance, by using a specific provision type oriented grammar (Fig. 1).

Some experiments testing the reliability of xmLegesExtractor have been carried out on a subset of 209 provisions. For

Class labels	Provision type	Dataset	Precision	Recall
c_2	Prohibition	13	85.71%	92.30%
c_3	Duty	59	69.23%	30.50%
c_4	Permission	15	78.95%	100.00%
c_5	Penalty	122	85.83%	89.34%
	Total	209	82.80%	73.68%

Table 4: **xmLegesExtractor experiments**

each class of provisions in the dataset the total number of semantic roles to be identified are collected in a gold standard dataset; this value was then compared with the number of semantic roles correctly identified by the system and the total number of answers given by the system. Some results are reported in Tab. 4.

[6]xmLegesExtractor has been developed in collaboration with the Institute of Computational Linguistics (ILC-CNR) in Pisa (Italy)

4.2. DK construction

Lexical units identified by xmLegesExtractor represent language-dependent lexicalizations of provision arguments. More information on related entities, as well as their relations within a specific domain, can be obtained by mapping lexical units to concepts in existing Domain Knowledges (DKs), if any. On the other hand the extracted information can be consider as a ground to construct DKs (in terms of thesauri or domain ontologies). Actually the construction of them is not a specific task of *legal* ontologists, but of ontologists *tout court*, since a DK has to contain information on entities of a domain independently from a legal perspective. This aspect is important in order to conceive a legal knowledge architecture whose components can be reused. A DILK-DK learning approach only suggests language-dependent lexical units for DKs, which can be implemented by projecting lexical units on a large text corpora of a specific domain, inferring conceptualizations by term clustering, as well as using statistics on recurrent patterns for discovering term relationships. This issue is out of the scope of this paper; a vast literature exists on this topic, therefore the interested reader can refer to (Buitelaar and Cimiano, 2008).

5. Conclusions

A knowledge modelling approach for the legal domain, called DILK-DK, has been presented. It aims to keep distinct domain knowledge from its legal perspective. Moreover an automatic approach based on machine learning and NLP techniques to support a bottom-up knowledge acquisition from legislative texts within the DILK-DK framework has been shown. The proposed learning approach for legal knowledge acquisition can provide the following benefits: a) it contributes to implement taxonomies or suggest concepts for hand-crafted ontologies (Walter and Pinkal, 2009); b) it contributes to bridge the gap between authoritativeness and consensus for legal rules representation, since it is able to extract rules directly form legislative texts, which are authoritative sources (by definition), nevertheless promoting consensus, since rules are automatically extracted from legal sources, limiting human interaction.

6. References

T. Agnoloni, L. Bacci, E. Francesconi, W. Peters, S. Montemagni, and G. Venturi. 2009. A two-level knowledge approach to support multilingual legislative drafting. In J. Breuker, P. Casanovas, M. Klein, and E. Francesconi, editors, *Law, Ontologies and the Semantic Web*, volume 188 of *Frontiers in Artificial Intelligence and Applications*, pages 177–198. IOS Press.

G. Antoniou, D. Billington, G. Governatori, and M.J. Maher. 1999. On the modeling and analysis of regulations. In *Proc. of the Australian Conference Information Systems*, pages 20–29.

L. Bacci, P. Spinosa, C. Marchetti, and R. Battistoni. 2009. Automatic mark-up of legislative documents and its application to parallel text generation. In N. Casellas, E. Francesconi, R. Hoekstra, and S. Montemagni, editors, *Proc. of the 3rd Workshop on Legal Ontologies and*

Artificial Intelligence Techniques joint with 2nd Workshop on Semantic Processing of Legal Texts, pages 45–54. Huygens Editorial.

R. Bartolini, A. Lenci, S. Montemagni, V. Pirrelli, and C. Soria. 2004. Automatic classification and analysis of provisions in italian legal texts: a case study. In *Proc. of the Second Int. Workshop on Regulatory Ontologies*.

J. Bentham and H. L. A. Hart. 1970 (1st ed. 1872). *Of Laws in General*. London: Athlone.

C. Biagioli and F. Turchi. 2005. Model and ontology based conceptual searching in legislative xml collections. In *Proc. of the Workshop on Legal Ontologies and Artificial Intelligence Techniques*, pages 83–89.

C. Biagioli, E. Francesconi, A. Passerini, S. Montemagni, and C. Soria. 2005. Automatic semantics extraction in law documents. In *Proc. of Int. Conference on Artificial Intelligence and Law*, pages 133–139.

C. Biagioli. 1997. Towards a legal rules functional microontology. In *Proc. of workshop LEGONT '97*.

J. Breuker and R. Hoekstra. 2004a. Core concepts of law: taking common-sense seriously. In *Proc. of Formal Ontologies in Information Systems*.

J. Breuker and R. Hoekstra. 2004b. Epistemology and ontology in core ontologies: Folaw and lricore, two core ontologies for law. In *Proc. of EKAW Workshop on Core ontologies*. CEUR.

J. Breuker, S. van de Ven, A. El Ali, M. Bron, S. Klarman, U. Milosevic, L. Wortel, and A. Forhecz. 2008. Developing harness. ESTRELLA Deliverable 4.6/3b, European Commission.

J. Breuker, P. Casanovas, M. Klein, and E. Francesconi, editors. 2009. *Law, Ontologies and the Semantic Web. Channelling the Legal Information Flood*, volume 188 of *Frontiers in Artificial Intelligence and Applications*. IOS Press.

C. Buckley and G. Salton. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.

P. Buitelaar and P. Cimiano, editors. 2008. *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, volume 167 of *Frontiers in Artificial Intelligence and Applications*. IOS Press.

P. Buitelaar, P. Cimiano, and B. Magnini. 2005. Ontology learning from text: an overview. In Buitelaar et al., editor, *Ontology Learning from Text: Methods, Evaluation and Applications*, volume 123 of *Frontiers in Artificial Intelligence and Applications*, pages 3–12. IOS Press.

T. Bylander and B. Chandrasekaran. 1987. Generic tasks for knowledge-based reasoning: the "right" level of abstraction for knowledge acquisition. *Int. Journal on Man-Mach. Stud.*, 26(2):231–243.

Nuria Casellas. 2008. *Modelling Legal Knowledge through Ontologies. OPJK: the Ontology of Professional Judicial Knowledge*. Ph.D. thesis, Institute of Law and Technology, Autonomous University of Barcelona.

B. Chandrasekaran. 1986. Generic tasks in knowledge-based reasoning: high-level building blocks for expert system design. *IEEE Expert*, 1(3):23–30.

W.J. Clancey. 1981. The epistemology of a rule-based expert system: a framework for explanation. Technical Report STAN-CS-81-896, Stanford University, Department of Computer Science.

E. Francesconi and A. Passerini. 2007. Automatic classification of provisions in legislative texts. *Int. Journal on Artificial Intelligence and Law*, 15(1):1–17.

A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider. 2002. Sweetening ontologies with dolce. In A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider, editors, *Proc. of the 13th Int. Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, volume 2473 of *Lecture Notes in Computer Science*.

T. Gruber. 2006. Where the social web meets the semantic web (keynote abstract). In I. F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, editors, *The Semantic Web - ISWC 2006, Proc. of the 5th Int. Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, page 994. Springer.

H. Hart. 1961. *The Concept of Law*. Clarendon Law Series. Oxford University Press.

R. Hoekstra, J. Breuker, M. Bello, and A. Boer. 2009. Lkif core : Principled ontology development for the legal domain. In J. Breuker, P. Casanovas, M. Klein, and E. Francesconi, editors, *Legal Ontologies and the Semantic Web*. IOS Press.

W. N. Hohfeld. 1978. *Some fundamental legal conceptions*. Greenwood Press.

H. Kelsen. 1991. *General Theory of Norms*. Clarendon Press, Oxford.

G. Lame. 2005. Using nlp techniques to identify legal ontology components: concepts and relations. *Lecture Notes in Computer Science*, 3369:169–184.

J.R. Quinlan. 1986. Inductive learning of decision trees. *Machine Learning*, 1:81–106.

J. Rawls. 1955. Two concepts of rule. *Philosophical Review*, 64:3–31.

J. Raz. 1980. *The Concept of a Legal System*. Oxford University Press.

A. Ross. 1968. *Directives and Norms*. London: Routledge.

J.R. Searle. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.

R. Studer, V. R. Benjamins, and D. Fensel. 1998. Knowledge engineering: Principle and methods. *Data Knowledge Engineering*, 25(1-2):161–197.

M. Uschold and M. Grüninger. 1996. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2):93–155.

S. Walter and M. Pinkal. 2009. Definitions in court decisions – automatic extraction and ontology acquisition. In J. Breuker, P. Casanovas, M. Klein, and E. Francesconi, editors, *Law, Ontologies and the Semantic Web*, volume 188 of *Frontiers in Artificial Intelligence and Applications*, pages 95–113. IOS Press.

Y. Yang and J.O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proc. of the Fourteenth Int. Conference on Machine Learning*, pages 412–420. Morgan Kaufmann Publishers Inc.