NLP Challenges for Eunomos, a Tool to Build and Manage Legal Knowledge

Guido Boella[1], Luigi di Caro[1], Llio Humphreys[1,2], Livio Robaldo[1], and Leendert van der Torre[2]

[1]Università di Torino, Italy. {guido, dicaro, humphreys, robaldo}@di.unito.it
[2]University of Luxembourg, Luxembourg. {llio.humphreys, leon.vandertorre}@uni.lu

Abstract

In this paper, we describe how NLP can semi-automate the construction and analysis of knowledge in Eunomos, a legal knowledge management service which enables users to view legislation from various sources and find the right definitions and explanations of legal concepts in a given context. NLP can semi-automate some routine tasks currently performed by knowledge engineers, such as classifying norms, or linking key terms within legislation to ontological concepts. This helps overcome the resource bottleneck problem of creating specialist knowledge management systems. While accuracy is of the utmost importance in the legal domain, and the information should be verified by domain experts as a matter of course, a semi-automated approach can result in considerable efficiency gains.

Keywords: ontology, legislative XML, classification, pattern matching, information extraction

1. Introduction

This paper discusses the use of natural language techniques for managing legislative information in Eunomos, a legal knowledge management service which enables users to view legislation from various sources and find the right definitions and explanations for legal concepts in a given context. NLP can semi-automate some routine tasks currently performed by knowledge engineers. Section 2. discusses the need for managing legislative information. Section 3. contextualises our research in semantic analysis of legislative text. Section 4. describes the use of statistical text classification to determine the topic of legislation at article level. Section 5. describes the use of statistical text similarity to find similar legislation that may be implicitly modified by new legislation. Section 6. describes the use of a parser and pattern matching rules to assist in the identification of references between legislations. Section 7. discusses the ontology and automated extraction of terms denoting concepts within legislation. Section 8. discusses extending the ontology to structure legal prescriptions, and using information extraction techniques to facilitate this process. Conclusion ends the paper.

2. Background

The growth of the internet means that it easier than ever to have access to the letter of the law. But as the law gets more complex, conflicting, and ever-changing, the law becomes incomprehensible to citizens and organisations and occasionally even to legal experts. Some of the most common problems are:

- laws coming from different sources;
- determining which laws override other laws;
- understanding legal terms and how they differ in meaning within different contexts, jurisdictions and over time;
- keeping up to date with legislative changes without being bombarded by irrelevant information.

Eunomos (Boella et al., 2011) is a legal knowledge management system developed as part of the ICT4LAW project, funded by Regione Piemonte, and distributed by Nomotika, a commercial spinoff of the Università di Torino. It was designed to have the following functionalities:

- the ability to view legislation at European, national and regional level from the same web interface, automatically downloading legislation from web portals.
- hypertext links between legislation that refer to other legislation;
- a list of similar legislation. This can help expert users find legislation that may have been implicitly modified or overridden;
- multilevel, update able ontologies so that users can see how terms are defined in different contexts, and track the evolution of terms over time;
- a mechanism for classifying norms in user defined categories such as taxation, immigration etc.;
- an alert message service to notify users when a newly downloaded legislation appears to be relevant to their domain of interest.

Eunomos is designed to be an online service provided to several clients, so that information and costs can be shared. Users can find the information they need quickly while the task of maintaining and updating information is left to knowledge engineers supported by automated technologies.

3. Semantic Analysis of Legislative Text

The text Eunomos downloads from legislative web portals are often in HTML or PDF formats. Eunomos uses the ITTIG CNR XMLLeges parser to transform such documents into XML documents in accordance with the NormaInRete standard. This means that every piece of legislation, and every section and article within it, are contained within XML tags with Uniform Resource Names (URNs), enabling cross-references to be immediately accessible via hyperlinks.

Class	Recall / with IG	Precision / with IG	F-Measure / with IG
C1	0.25 / 0.22	0.32 / 0.33	0.28 / 0.26
C2	0.71 / **0.80**	0.64 / **0.67**	0.67 / **0.73**
C3	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
C4	0.64 / **0.72**	0.80 / **0.90**	0.71 / **0.80**
C5	0.95 / **0.97**	0.87 / **0.92**	0.91 / **0.95**
C6	0.40 / **0.80**	0.67 / **0.80**	0.50 / **0.80**
Weighted average	0.71 / **0.76**	0.69 / **0.74**	0.70 / **0.74**

Table 1: Classification results after the Information Gain-based feature selection step, using Support Vector Machines (SVM) and 10-fold cross validation. The improvements in accuracy are highlighted in bold. The only class that decreases in performances is C1, whereas class C3 remains completely misclassified because of the scarcity of its data (it only contains three documents).

Structural analysis of legislative text is an essential starting point from which to begin semantic analysis, and natural language techniques to address the following:-

- what is the topic of this legislation and articles within it (text classification);
- what other legislation are related to this legislation (text similarity);
- in what way are these legislations related (pattern matching);
- what do these terms mean in this context (ontologies);
- what are the legal prescriptions arising from this legislation (information extraction).

4. Determining Topic of Legislative Articles Using Text Classification

For each new piece of legislation, the classification task is:

1. to find which domains are relevant to the legislation, and
2. to identify which domain each article belongs to (in Italy, there are laws that contain norms about different and unrelated domains).

The first task enables targeted email notification messages to be sent to all users interested in the particular domains covered by new legislation. The second task enables users to view, in each piece of new legislation, only articles relevant to a particular domain. If at a first sight this seems a straightforward application of standard classification techniques, the scenario provided by Eunomos offers some original aspects due to the peculiarities of legislation and of the work of the legal knowledge engineer. Our hypothesis was that legal text is very similar in style and content such that it is difficult to distinguish between one topic and another; nevertheless, it might be possible to use semi-supervised classification to help the work of a knowledge engineer.

Text classification becomes plausible the moment the legal inventory of Eunomos is sufficiently populated with articles that have already been classified. Since the construction of the inventory is still under way, focusing mostly on financial regulations and health & safety, we conducted experiments using the classification of legislation offered on the portal of Regione Piemonte's tax office[1].

Although there are plenty of algorithms for text classification, we used the well-known Support Vector Machines (SVM) for this task, since it frequently achieves state-of-the-art accuracy levels (Cortes and Vapnik, 1995; Joachims, 1998). This algorithm represents a text document as a vector of terms frequencies (Salton et al., 1975), allowing direct and efficient comparisons of documents by measuring the deviation of angles between their vectors (Manning et al., 2008). More in detail, we used the Sequential Minimal Optimization algorithm (SMO) (Platt and others, 1999) with a polynomial kernel. The vectorial representation of textual data is particularly complex as it usually contains thousands of features. One property of SVM is its ability to learn from cases with high dimensionality of the feature space (Joachims, 1998), so it fits well with text.

The process of transforming text into vectors requires selection of suitable terms, and use of a weighting function as part of the frequency calculations. We used the *Term Frequency-Inverse Document Frequency* (TF-IDF) weighting function as proposed in (Salton and Buckley, 1988), that takes into account both the frequency of a term in a text and how characteristic it is of text belonging to a particular class. There are pre-processing steps that can be carried out on the selection and transformation of terms, which have been shown to be more effective than a simple bag-of-words approach. A commonly-accepted technique is to use a stopword list to remove uninformative terms and morphological transformation of remaining terms to to their lexical roots (i.e., the lemmas). The aim of these procedures is to eliminate noise while reducing redundant linguistic variability. Typically only nouns are then considered as informative features. The accuracy of the classification methods is highly dependent on the quality of these procedures. Our approach differs from standard practice of using lists of stopwords and external resources such as WordNet (Miller, 1995) to extract the lemmas, in that we use a dependency parser for Italian called TULE (Lesmo, 2009) that performs a deep analysis over the syntactic structure of the sentences.

[1] http://www.regione.piemonte.it/

The use of a syntactic parser is the basis for fine-grained pre-processing:

1. it allows the disambiguation of terms at syntactic level;
2. it allows for direct selection of the informative units (i.e., lemmatized nouns, verbs and so on);
3. it can be used for the generation of *semantic chunks*, i.e., syntactic subtrees that express high-level concepts. In our future work we intend to study this type of pattern to see if can be used to outperform our current classification tool.

Our approach may increase the system complexity as a whole, but it is a better solution for this domain than WordNet-like methods which only have top-domain ontologies and thus are unable to recognise and lemmatize many legal domain-specific terms.
The association between features such as words from legislation articles and a category label was fed to an external application based on the WEKA toolkit (Hall et al., 2009) and incorporated in Eunomos. The WEKA toolkit (Hall et al., 2009) was used as a framework for the experiments because it supports several algorithms and validation schemes allowing an efficient and centralized way to conduct experiments and evaluate the results. The SVM classifier achieves an accuracy of 71% when trained with the n-folds cross validation scheme (Kohavi, 1995) (using n = 10, which is a common practice in the literature). Indeed, the classifier achieves an accuracy of only 90% when trained and tested on the same dataset.
In order to improve the performance of the classifier, we applied further machine learning techniques to evaluate the importance of the features in terms of their discriminatory power among the classes. Information Gain (IG) is a popular way to capture this knowledge (Yang and Pedersen, 1997). Generally speaking, Information Gain (IG) is a measure that calculates the entropy between two probability distributions. Once such values are obtained, it is possible to rank the features and prune them according to a threshold. In our case, it is used for removing those terms (i.e., seen as probability distributions) that do not carry any information for the estimation of the class of the instances (i.e., the probability distribution of the class). Table 1 shows an improvement when using this technique with a threshold of 0.0, i.e. only the features with positive IG were kept. While the SI grows to 74.45% (+4.5%), the general accuracy of the classifier increases from 70.85% to 76.23%, and all the classes are better generalised apart from class C1 where it decreases slightly. This confirms Greece's (Greene, 2001) claim that SI is positively correlated with the accuracy of an SVM classifier. Finally, class C3 remains misclassified because of data scarcity.
In (Boella et al., 2012a), the classification framework is extended to consider as features also the ontological relations between the concepts mentioned in the law.

5. Finding Related Legislation Using Text Similarity

For each piece of legislation, Eunomos generates a list of the most similar pieces of legislation in the whole database using Cosine Similarity. This list is useful for finding related or implicitly modified legislation when a new piece of legislation comes in. The Cosine Similarity is also used by knowledge engineers in the bootstrap phase of a domain. If a client is interested in a new domain, e.g. privacy, well-known legislation for that domain can be used to retrieve similar norms in other legislation in the Eunomos repository, which can then be classified as belonging to the same domain. Where labelled data is not available, Cosine Similarity is a precondition to build the training set for the classifier described in the next section.

Applying Cosine Similarity to search for relevant text is a common practice in general-purpose Information Retrieval tasks. In these cases, the main issue is to determine how many items to select and return. This means choosing an appropriate threshold (or cutoff) to apply to the ordered list of relevant articles created with the Cosine Similarity measure.

A naive solution for truncating the list of articles that are ordered by its similarity with the input one is to use a fixed cutoff k. This way, only the first k articles have to be considered as relevant. However, this approach does not take into account the distribution of the ordered similarity values. An alternative approach is to find where the similarity values suffer a *significant fall*. This separates the truly similar articles from the rest. A practical way to implement this idea is to analyse the distribution of the ordered values looking at the highest difference (or highest "jump") between adjacent values in the list, as done in (Cataldi et al., 2009b).

In our experiments, we made use of the topics associated with the articles (see Section 4.) as part of the evaluation process. More in detail, given one article a and a set of similar ones S_a, the evaluation task looks at whether the articles contained in S_a have the same topic as the input article a. Figure 1 shows the result of the accuracy when fixing the cutoff k, and when using such article-level automatic estimation of k. This shows that, notwithstanding the benefit of using a variable and data-dependent approach for estimating the cutoff k, the accuracy level reached by this technique is noticeably higher than with the use of fixed cutoffs.

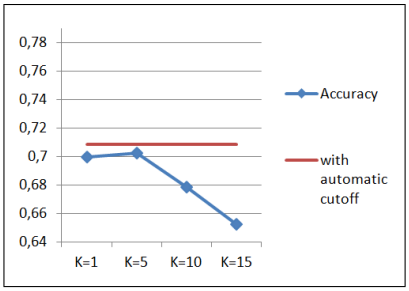

Figure 1: Evaluation of the accuracy of the Cosine Similarity-based approach for finding relevant articles, using the class labels associated to the articles. Note that the accuracy levels reached by the automatic technique is higher than with the use of fixed cutoffs.

6. Finding Reference Types Using Pattern Matching

Currently, the XMLLeges parser can find most explicit references but it cannot annotate the reference type as specified by the *NormeInRete* standard (or NIR) for Italian Legal Text. The NIR standard defines some structural elements that are used to mark up the main elements of a legal text, as well as its atomic parts (such as articles, paragraphs, *etc.*) and any non-structured text fragment.

A provision can be encoded through a specially defined space called <*meta*>, in which a *URN* connects the element expressing a qualification with the textual element referred to (be this an atomic element, or a text string). We report here the modificatory provision model and definitions, presented in (Palmirani and Brighi, 2006), on which our research is based. A *modificatory clause* includes the following information:

- *ActiveNorm*: the provision that states the normative modification;
- *PassiveNorm*: the provision that is affected by the modification;
- *Action*: action produced by the active provision on the passive norm;
- *Times*: efficacy of the modificatory provision;
- *Content*: the old text to replace or repeal in the modified provision, as well as the new text is inserted in the destination;
- *Purview*: a part used to describe a modification, as by specifying any exception, extensions, or authorized interpretations;
- *Space*: a function used to specify a geographical area to which the modification applies;
- *Conditions*: where a modification is an effect dependent on an event, a geographic space, or a class (or domain) of application.

Currently, a knowledge engineer manually specifies whether the reference is a simple reference or it modifies or overrides other legislation and extracts all the elements of the modification, such as date, target, etc. But we are looking to use rule-based pattern matching technologies to semi-automate this process.

The rules are implemented in XML, but space constraints prevent us from illustrating in detail the XML format adopted. The general pattern of the rules is illustrated in Figure 2.

The system scans the words in the input text and, in case it finds a word with lemma *K1*, it triggers the rule in Figure 2. Then it carries out three checks.

Firstly, it checks if the morphological information of the keyword with lemma *K1* matches the one in $Morph_{K1}$. Then it checks whether the words that follow the keyword match the morphological descriptions precisely and in the same order $Morph_{W_{n1}}, \ldots, Morph_{W_{nx}}$ and whether the words that precede it similarly match the morphological descriptions $Morph_{W_{p1}}, \ldots, Morph_{W_{py}}$.

$dist_{n1}, \ldots, dist_{nx}, dist_{p1}, \ldots, dist_{py}$ are integers specifying the maximal distance among a pair of words. For instance, between the keyword and the word W_{n1} there could be $dist_{n1}$ other words.

If the three checks are satisfied, the provision is classified as type *T1*. Moreover, among the words W_{n1}, ..., W_{nx}, W_{p1}, ..., W_{py}, there could be some normative references, that could be classified by the rule as either *norma* (norm), *novella* (replacement text), or *posizione* (position). In case the constraints specified in the rule are satisfied, the final annotation will specify the normative references recognized by it.

Figure 3 shows an example of instantiation of the pattern in Figure 2. The rule is triggered when the system finds in the input text a verb with the lemma 'sopprimere' (*to suppress*). Then, it checks whether there is a verb with lemma 'essere' (*to be*) among the two[2] preceding words, and whether there is a normative reference among the five preceding words of the lemma 'essere'. Where this is the case, the provision is annotated as 'abrogazione', with the normative reference occurring therein identified as 'norma'.

Many provisions are correctly classified by the rule in Figure 3. Nevertheless, the rule can also lead to wrong annotations. Although the main verb of some provisions is 'abrogare', the text is technically a 'sostituzione'. Generally, sentences of the form 'Il rif1 è abrogato da rif2' (*The rif1 is abrogated by rif2*) are substitutions, not abrogations. Therefore, we add in the system the rule in Figure 4, and of course assign it a higher priority than the rule in Figure 3.

The checks carried out on the words preceding the keyword 'abrogare' are the same as for those in Figure 3. Furthermore, the rule in Figure 4 requires the occurrence of the preposition 'da' immediately after the keyword and a normative reference (that will be annotated as 'novella') among the five words following the preposition.

Our system was inspired by the work of (Lesmo et al., 2009), and evaluated on the same test set.

Our system is much simpler from a computational perspective than the multi-layered architecture of (Lesmo et al., 2009). This makes it faster to run, and easier to handle and tune incrementally. One problem with the system of (Lesmo et al., 2009) is that the errors caused by the TULE parser, which is a *multi-purpose* parser, propagate in the final result. Our system achieves a higher level of precision, close to 100%, because the rules behave as a kind of "filter". In other words, the system uses *custom* rules, each of them describing a valid pattern. As a consequence, (almost) any provision matching with these patterns are precisely accurately. On the other hand, the updating process, i.e. the task of adding new rules to the system, is slow and tedious.

The results of both systems are shown in Table 2. However, a deeper analysis is needed to fully assess the system

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

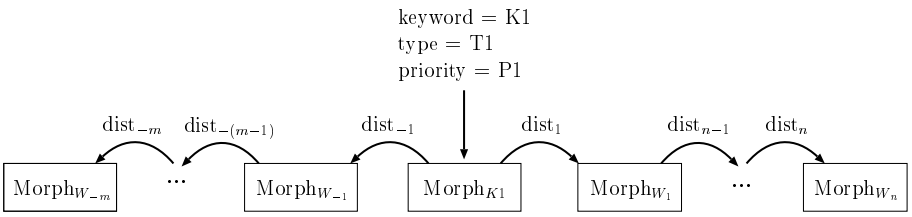

Figure 2: General pattern of the system rules

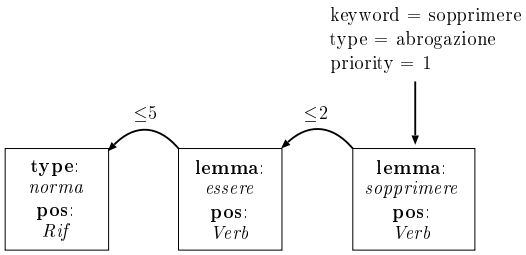

Figure 3: A rule for some kinds of 'abrogazioni' (*abrogations*)

and to compare our pattern-matching approach to the multi-layered approach.

7. Finding the Right Meaning of Legal Terms Using Multi-level Ontologies

Legal text can be difficult to understand because each term, each concept, has a strict and defined but sometimes obscure meaning. Sometimes these terms are defined in the same piece of legislation. Sometimes their meaning can be found in other legislation or even in judicial or scholarly interpretations. To aid understanding of legal terms, the Eunomos package incorporates the Legal Taxonomy Syllabus, a specialist multilevel multilingual ontology. In the Legal Taxonomy Syllabus, to properly manage terminological and conceptual misalignment, a distinction is made between *legal terms* and *legal concepts*. The basic idea in the system is that the conceptual backbone consists in a taxonomy of unique concepts (ontology) to which any number of terms can refer to express their meaning. The Legal Taxonomy Syllabus ontology stores concepts and terms in separate database tables.
Each concept in the terminology has the following fields:

- language
- jurisdiction
- domain
- description in natural language
- references to relevant articles
- notes
- links to related concepts

Building ontologies is a resource intensive task, so from the Eunomos interface, new terms and interpretations can be added to the ontology directly from definitions in the text of the law. This also ensures that the construction of concepts is strictly integrated with the norms defining them. Eunomos can find exact matches of known terms and highlight them in the text of the law. We are now working on semi-automating the process of linking terms in the text of the law to concepts in the ontology using natural language processing techniques. The first phase in this work package is to semi-automate the process of identifying and classifying known terms. The tool uses the TULE part-of-speech (POS) tagger and parser (Lesmo, 2007) to identify all nouns and verbs as well as compound nouns such as 'bank director' in the legislation, and searches for these terms in the relevant ontology. Since each article is classified as belonging to a certain domain, the tool looks only for terms within the appropriate domain-specific ontology, thereby helping to avoid the problem of multiple conceptualisations for terms. The system also identifies variations of terms, e.g. 'bank director' and 'director of a bank', by searching for words with the same lemmas in the TULE dictionary, and using a custom pattern matcher developed for this purpose. The task of linking terms in legislation with their definitions in the ontology is a semi-automated task. A knowledge engineer needs to check the proposed links, looking out for POS tagging errors, such as 'rischio' tagged as a verb instead of a noun. Nevertheless, such errors are relatively rare, and the pattern matcher can easily be updated to handle common tagging errors. Preliminary experiments have shown a very high level of recall and precision.

The next work in this area will look at automated means of adding terms and concepts to the ontology. To date, the Eunomos ontology includes 500 terms related to compliance, since there has been much interest in using the Eunomos within this sector. To extend the coverage, we are considering adding terms and concepts from generic legal ontologies such as LOIS, JurisWordNet or core legal ontologies by extracting the ontologies to RDF/OWL. To extract terms from a new specialist area of law we could integrate an unsupervised technique called TMine (Candan et al., 2008), which is able to automatically bootstrap a domain ontology from a set of plain texts making use of statistical tech-

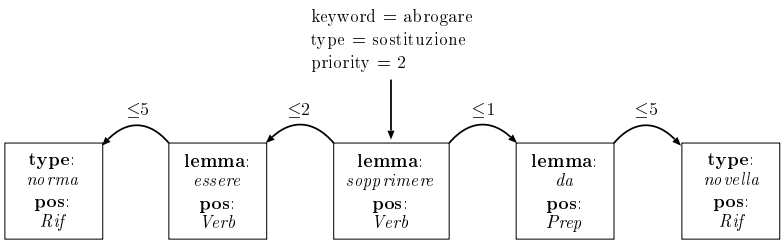

Figure 4: A rule for certain kind of 'sostituzioni' (*substitutions*)

	Recall	Precision
(Lesmo et al., 2009)'s System	71.7%	83.0%
Current System	86.60	98.56

Table 2: Evaluation of the system, and comparison with (Lesmo et al., 2009)

niques such as Latent Semantic Analysis. This can help both with exploration of the data and creation of initial categorizations to be verified by experts. We wish to improve this technique by considering entire noun phrases as well as single terms. Meanwhile, Cosena (Cataldi et al., 2009a), could enrich the ontology by finding terms that correspond to other terms in usage, and that can be mapped to existing concepts. Given a concept, it measures the frequency of terms with the same contexts together with their frequency out of such contexts. An added benefit of this process is that it is effective at finding new terms that are candidates for being new concepts.

Our future work will look at the use of semantic technologies to map prescriptions to Business Process Management (BPM) activities (e.g., in-house banking processes). Banks manage thousands of BPM activities and this new component is the next step in ensuring that these banking processes are compliant. Here, the challenge for NLP is to align the terminology used in the law and the one used in the description of the processes.

8. Finding Legal Obligations Using Information Extraction and Structured Data

The basic Eunomos system described above has been extended in (Boella et al., 2012b) so that the ontology includes prescriptions on what actors must do or must not do to comply with the law. This feature is useful for financial institutions who are subject to complex regulations that are updated frequently. Within Eunomos, each prescription contains the following fields:

- deontic clause : the type of the prescription: obligation, prohibition, permission, exception;
- active role : a concept subsumed by the concept role (e.g., citizen, director) which is the addressee of the norm;
- passive role : the beneficiary of the norm;
- crime : a concept in the ontology of crimes resulting from the violation of the prescription (if it is an obligation or prohibition). This concept is often defined in other legislation;
- sanction : a concept describing the sanction resulting from the violation. All elements are linked to definitions within legislation such as the Italian Penal Code via URN.

The work of extracting legal obligations from laws and populating the prescriptions ontology is currently carried out entirely by legal experts. Our future work will involve the use of automated information extraction (IE). Information extraction from natural text is challenging because of language variability: the fact that the same information can be expressed with different words and syntactic constructs. There are further factors to be taken into consideration when processing information legislative text:

- long sentences with several clause dependencies
- lists, where each item are usually not standalone sentences
- references to other articles, the content of which is not quoted within the referring article.
- difficulties for inter and intra-sentential anaphora resolution

Legal text is an under-researched area in IE (but see e.g., (Biagioli et al., 2005)), and there is a lack of suitable annotated data. We are therefore looking at unsupervised techniques. For example, (Szpektor et al., 2004)'s TEASE system is a paraphrasing extraction system that extracts relations between a pivot (lexical entry) and a template (dependency parse fragment). Syntactic word order patterns, such as active/passive formulations can be generated according to standard template rules and grouped together in equivalence classes.

9. Conclusion

Information technology is a natural ally for legal research, characterised as it is by constant cross-referencing, updates and obscure terminology. Natural language processing is essential for efficient semantic analysis of legislative text.

In this paper we illustrate ongoing work on the Eunomos software, developed to support the work of legal professionals by offering them an environment which makes laws easier to navigate, annotate and understand. We have described the key functionalities of the system, and ways in which natural language processing techniques can be used for specific tasks in knowledge management within the system.

10. References

C. Biagioli, E. Francesconi, A. Passerini, S. Montemagni, and C. Soria. 2005. Automatic semantics extraction in law documents. In *The Tenth International Conference on Artificial Intelligence and Law (ICAIL)*, pages 133–140.

G. Boella, L. Humphreys, M. Martin, P. Rossi, and L. van der Torre. 2011. Eunomos, a legal document and knowledge management system to build legal services. In *AI Approaches to the Complexity of Legal Systems (AICOL11)*.

G. Boella, L. di Caro, L. Humphreys, and L. Robaldo. 2012a. Using legal ontology to improve classification in the eunomos legal document and knowledge management system. In *Semantic Processing of Legal Texts Workshop (SPLeT 2012) at LREC 2012*.

G. Boella, M. Martin, P. Rossi, L. van der Torre, and A. Violato. 2012b. Eunomos, a legal document and knowledge management system for regulatory compliance. In *Proceedings of Information Systems: a crossroads for Organization, Management, Accounting and Engineering (ITAIS) Conference*, Berlin. Springer.

K. Selçuk Candan, Luigi Di Caro, and Maria Luisa Sapino. 2008. Creating tag hierarchies for effective navigation in social media. In *Proceeding of the 2008 ACM Workshop on Search in Social Media, SSM 2008*, pages 75–82.

M. Cataldi, C. Schifanella, K. Selçuk Candan, M.L. Sapino, and L. Di Caro. 2009a. Cosena: a context-based search and navigation system. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, MEDES '09, pages 33:218–33:225, New York, NY, USA. ACM.

M. Cataldi, C. Schifanella, K.S. Candan, M.L. Sapino, and L. Di Caro. 2009b. Cosena: a context-based search and navigation system. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, page 33. ACM.

C. Cortes and V. Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

J. Greene. 2001. Feature subset selection using thornton's separability index and its applicability to a number of sparse proximity-based classifiers. In *Proceedings of Annual Symposium of the Pattern Recognition Association of South Africa*.

M. Hall, E. Frank, G. Holmes, F. Pfahringer, P. Reutemann, and I.H. Witten. 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11:10–18, November.

T. Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98*, pages 137–142.

R. Kohavi. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International joint Conference on artificial intelligence*, volume 14, pages 1137–1145.

L. Lesmo, A. Mazzei, and D.P. Radicioni. 2009. Extracting Semantic Annotations from Legal Texts. In *Procs. of HT09*, pages 167–172, Turin, Italy, July. ACM.

L. Lesmo. 2007. The Rule-Based Parser of the NLP Group of the University of Torino. *Intelligenza Artificiale*, 2(4):46–47, June.

L. Lesmo. 2009. The Turin University Parser at Evalita 2009. *Proceedings of EVALITA*, 9.

C.D. Manning, P. Raghavan, and H. Schutze. 2008. *Introduction to information retrieval*. Cambridge University Press.

G.A. Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

M. Palmirani and R. Brighi. 2006. Time Model for Managing the Dynamic of Normative System. *Electronic Government*, pages 207–218.

J. Platt et al. 1999. Sequential minimal optimization: A fast algorithm for training support vector machines. *Advances in Kernel Methods-Support Vector Learning*, 208:98–112.

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

G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM*, 18:613–620, November.

I. Szpektor, H. Tanev, I. Dagan, and B. Coppola. 2004. Scaling web-based acquisition of entailment relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004*, pages 41–48.

Y. Yang and J.O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Machine Learning International Workshop*, pages 412–420.