Mission Impossible?: Automated Norm Analysis of Legal Texts

Emile de Maat Tom M. van Engers
Leibniz Center for Law
Faculty of Law, Universiteit van Amsterdam
Amsterdam, The Netherlands
{*demaat, vanengers*}*@lri.jur.uva.nl*

1 Introduction

Although many legal experts thought this would be impossible we are very close in creating an algorithm for automated norm analysis from legal texts. This algorithm makes use of invariant linguistic structures at the syntactical level that characterises specific normative expressions in natural language. Since the algorithm has not been realised and tested completely jet, we will limit ourselves in this article to explaining the invariances in the natural language representations in which norms in legal texts are expressed.

As part of the POWER research program [1], the Dutch Tax and Customs Administration has created a method to formalise normative expressions in legal texts in UML/OCL models. These UML/OCL representations of the legal texts have showed to be quite suitable. To support knowledge analysts in creating these UML/OCL-models, an automated concept extractor was created, which allows a computer to identify the different concepts that exist in a given legal text [2]. This automated concept extractor reduces the amount of work of the knowledge analysts and results in more uniform models as well. The research described here is aimed at further automating the translation of a legal text to a model.

Automated generation of models would not only lead to a reduction in the amount of work needed, it would also increase inter-analyst independency. Normally, models created by different analysts could differ in various small details. Removing those difference would lead to more uniform models, which can more easily be understood, and are also easier to process when creating applications based on these models.

This article discusses the first results of this research into automated analysis of legal texts.

2 Main Sentences in Legal Texts

Research within the POWER project has lead to the conclusion that the sentences that occur in legal texts can be grouped in few categories. These main categories are: *definitions and type extensions, deeming provisions, application provisions, value assignments and changes, norms*. For each category (with the exception of norms), there is a limited set of possible language constructs used in the sentence. A sentence can be classified by studying the construct used.

The main sentences can be extended by adding a subordinate clause. Just like the main sentences, these extensions can also be identified by the language constructs used in the subordinate clauses.

3 Applying Sentence Categories: Parsing

An important advantage of the sentence types defined above is that they make it possible for a computer to classify the different sentences. Because all but one of the categories can be identified by the language constructs that are used, a computer can test for the presence of these constructs.

In order to be able to parse sentences, we will need to formalise the patterns that can occur. In order to identify the sentence types, this could be done by building a context-free grammar. Normally, context-free grammars are not the most successful tool to model natural languages. This is because natural languages are highly ambiguous and context-sensitive [3]. Normally, a problem arises when we try to parse a sentence like "The man walked to the woman with a limp." This sentence is ambiguous, since there are two different interpretations: the man has a limp, or the woman has a limp. It is not possible to distinguish between the two different interpretations, unless we have got more information.

However, we will be able to use the context-free grammars to parse and translate legal sources. Syntactical ambiguity should not be a problem here: the text should be syntactical unambiguous[1], since this is a legal text, a law. We do not want any lawyers arguing that the sentence "You are not allowed to shoot anyone with a gun." means that you can shoot anyone who doesn't carry a gun. Thus that problem is not present here. Of course, although a legal source should be syntactical unambiguous, there can still be ambiguity in a given legal source, which cannot be handled correctly. However, since it is an error in the legal source (it should be unambiguous), it is not a problem if it is not translated correctly.

Finally, it is not necessary to recognize the entire sentence; it is sufficient to parse the top levels. This is because (in the POWER project) great parts of the sentences are kept in one piece. For example, a deeming statement has the following pattern: "*noun-phrase* [wordt geacht] *fiction*". When building a model of this text, the entire fiction (the largest part of the sentence) is often kept in one piece.

4 Conclusion

Legal texts, although expressed in natural language, provide us with enough clues to identify norms merely by looking at the syntactical structure. This allows us to develop an algorithm for automated norm extraction. We hope to have the first prototype ready in the beginning of 2004. Although we don't claim 100% recognition, a significant reduction of knowledge analysis effort (and improvements realised in inter-coder independencies) will mean a small revolution in the AI& Law world.

References

[1] Van Engers, T.M., Boekenoogen, 2003, Improving Legal Quality - an application report, in Proceedings of ICAIL2003, ISBN 1-58113-747-8, ACM Press.

[2] R. van Gog and T.M. van Engers, Modelling Legislation Using Natural Language, *Proceedings of the 2001 IEEE Systems, Man and Cybernetics Conference.*

[3] H. Alblas and A.Nymeyer. Practices and Principles of Compiler Building with C. Prentice Hall, London, 1996.

[1] Note that the legal text should only be syntactical unambiguous. This does not mean that the legal text should be entirely unambiguous. It will always retain a certain degree of vagueness. However, this ambiguity is semantically, and lies in the interpretation of the words, not in the interpretation of the structure of the sentence. This does not affect the translation process, since the words are retained in the model.