Linguistic Description and Automatic Extraction of Definitions from German Court Decisions

Stephan Walter
Department of Computational Linguistics, Saarland University, Saarbrücken
Building C 72, PO Box 15 11 50
E-mail: stwa@coli.uni-saarland.de

Abstract

This paper discusses the use of computational linguistic technology to extract definitions from a large corpus of German court decisions. We present a corpus-based survey of definition structures used in this kind of text. We then evaluate the results of a definition extraction system that uses patterns identified in this survey to extract from dependency parsed text. We show how an automatically induced ranking function improves the quality of the search results of this system, and we discuss methods for the acquisition of further extraction rules.

1. Definitions in Court Decisions

Besides normative content, the statutes of code law systems comprise terminological knowledge. This terminological knowledge consists in definitions of concepts used to describe the facts sanctioned by the law. Article 1 of the German Federal Water Act e.g. captures a specific terminological sense of *waters* as follows:

(1) Dieses Gesetz gilt für folgende Gewässer:
1. das ständig oder zeitweilig in Betten fließende oder stehende (..) Wasser
[This Act shall apply to the following waters:
1. permanently or temporarily flowing or standing (…) waters confined within a bed]

If the definitions contained in statutes would fully specify how the relevant concepts are to be applied, cases could be solved (once the relevant statutes have been identified) by mechanically checking which of some given concepts apply, and then deriving the appropriate legal consequences in a logical conclusion.
However, discussion in courts (and consequently texts that document court decisions) is largely devoted to pinning down whether certain concepts are to be applied or not. Many evaluative concepts such as e.g. *significant value*, cannot be captured by general descriptive definitions at all. However even relatively concrete descriptive concepts, such as *waters* in (1) often need to be supported with further definitions in courts' decisions. The definitions in (2) are quoted from a decision by the Higher Administrative Court of Hamburg. Sentence (2.1) fixes what generally counts as the bed of a body of water (a concept that is used in (1) but not defined in the Federal Water Act), and (2.2) states precisely how this definition is to be applied regarding the specific case of tubed ditches.

(2.1) Unter einem Gewässerbett ist eine äußerlich erkennbare natürliche oder künstliche Begrenzung des Wassers in einer Eintiefung an der Erdoberfläche zu verstehen (vgl. BVerwG, Urt. v. 31.10.1975, BVerwGE Bd. 49 S. 293, 298; Beschl. v. 17.2.1969, Buchholz 445.4 § 1 WHG Nr. 3, m.w.N.).
[By a bed of a body of water is to be understood: the natural (…) confines of water within a cavity in the surface of the earth. (cf. BVerwG, Urt. v. 31.10.1975, BVerwGE Bd. 49 S. 293, 298; Beschl. v. 17.2.1969, Buchholz 445.4 § 1 WHG Nr. 3, m.w.N.).]

(2.2) Von einem derartigen Bett kann u.a. dann nicht mehr gesprochen werden, wenn ein Graben vollständig verrohrt wird.
[Such a bed of a body of water (…) can no longer be assumed if a ditch is fully tubed.]

On the one hand - as can be seen from the extensive amount of citation sources mentioned in brackets in (2.1) – such definitions are frequently re-used and remain binding beyond the case at hand. On the other hand they are generally open for later amendment and modification. The semantics of legal concepts is thus subject to constant adaptation and revision within use (Hart (1961) uses the expression *open texture* to characterize this fundamental property of law).

Legal methodology imposes high standards on the explicitness and transparency of this adaptation process, and it is of great interest from the point of view of linguistics as well as legal theory to learn how judges try (successfully or not) to comply with these standards. Section 3 of this paper contributes to this aim. It provides a framework for describing the linguistic means used to express legal definitions and applies it in a corpus study based on 100 German verdicts.
Moreover, access to judges' definitions is of great importance to the legal practitioner. Judges need to know

such definitions in order to achieve a uniform application of the law over longer periods of time, and lawyers may be provided with valuable arguments to make their clients' case. Sections 4 and 5 of this paper discuss the design, implementation and evaluation of a legal definition extraction system that based on the linguistic realization patterns described in Section 3. The system accesses a corpus of more than 35 000 German court decisions (~ 75 million tokens).

2. Related Work

Definitions have been studied in philosophy since the works of Plato and Aristotle. Yet the question of how definitions are realized linguistically has only been investigated closer in modern applied linguistics, in particular in terminology research and in lexicography. In these areas however the topic has mainly been looked at from a prescriptive point of view (e.g. Landau, 1984; Arntz & Picht, 1989; ISO 860 and 704).

Research on advanced information technology in the legal domain has up to now mostly been concerned with legal reasoning and knowledge representation. One focus of interest has been legal ontologies (cf. Valente, 2005). There has been only little research on the use of natural language processing in this context (Lame 2005; Saias & Quaresma, 2005, and – to some degree – the European LOIS project, Dini et al., 2005).

Automatic identification of definitional information has mainly been studied for definitional question answering at the TREC-competitions (e.g. Peng et al., 2005; Hildebrandt, Katz & Lin, 2004; Lin & Demner-Fushman, 2005). Outside q & a, direct definition identification has been investigated for the domains of English technical documentation (e.g. Meyer, 2001 and Pearson 1998; Storrer & Wellinghoff, 2006 work with German technical text) and biomedical text (Klavans & Muresan, 2001; Fahmi & Bouma, 2006).

3. Linguistic description of definition types

The prototypical structure dealt with in most literature on definitions looks as follows (so called 'Aristotelian' definition scheme):

An A is a B which C

This construction also occurs in court decisions, but there it is by no means the only way used to make defining statements. Firstly, the relation between the defined term and the defining phrase can be established by means of appositive constructions instead of clause predicates. Through appositive constructions, either the defined term or the defining phrase is marked as background information. Such constructions therefore often serve to remind the reader of definitions that are presupposed or have been given explicitly somewhere else in the text.

Moreover, a considerable number of different predicates can be used in clause-based definitions apart from copular *be*. One important reason for this high degree of variability in formulation lies in the specific role of definitions in court decisions. Scientific and technical terminology is often built up using more or less context-free general definitions assigning new terms to places within a given taxonomy. In contrast, defining statements in verdicts are parts of coherent texts and do not only serve as specifications of terms, but also as arguments for or against their application in a specific case. Example (2) exhibits a typical macro-structure of such 'definitional arguments': A general core definition (2.1) is elaborated by an additional, more specific statement (2.2) serving an argumentative goal with respect to the given case. Such elaborations may be given dialectically, for and against the use of the concept (here: against the classification as bed of water). Additionally, background information – not conclusively supporting, but 'pointing to' the (non-)applicability of the respective concept – may be attached to each of the elaborating statements in such a sequence.

To arrive at an overview of the distribution and relative importance of the various types of legal definition statements, we performed an exploratory analysis of a corpus of 40 German court decisions (comprising 127 349 tokens in 3757 sentences, "pilot study corpus"). Based on the results of this study we devised guidelines for a controlled double annotation of another sixty decision texts (233 210 tokens in 7627 sentences) from which we then constructed a gold standard resource for the extraction experiments described in Sections 4 and 5 ("gold standard corpus").

The documents analysed in our pilot study contain a total of 126 definitions. 36 of these were realized by appositive means, 90 include at least one predicate-based core-statement, with an additional 18 elaboration and 21 background statements (1.5 sentences per definition on average). These definitions use 52 different predicates that mostly fall into the following four classes:

1. expressing a classification: e.g. copular *sein-be*, classificatory verbs such as *fallen unter- fall under* or (less neutrally) *gelten als – be considered as*
2. meta-linguistic, used to speak directly either about word meaning or conditions of applicability (e.g. *bedeuten – mean* or *vorliegen – 'be existent'*)
3. referring to aspects important to the process of legal interpretation (e.g. *fordern – require*, *darstellen - constitute*)
4. naming a specific type of feature used in the respective definition (e.g. *dienen zu – serve as*, *schützen - protect*)

The diagram in Figure 1 shows that the relatively general classificatory and meta-linguistic predicate types are most frequent in core statements. Within the two types of additional statements, legal interpretational questions are at issue. This is reflected by the relative prominence of the class of interpretation related predicates. Finally, background statements tend to give quite specific

information on the defined concept, and therefore contain feature specific predicates more often than the other two statement types.

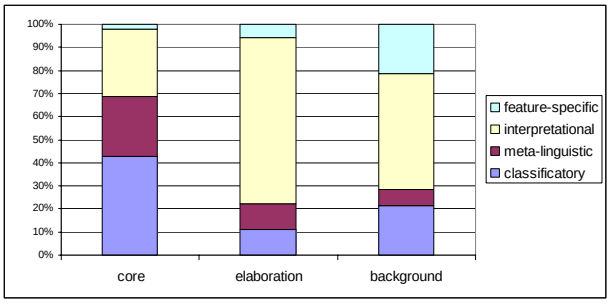

Figure 1: Distribution of predicate classes in sentence types

The controlled double annotation of the gold standard corpus was performed by one graduate law student and one post-graduate computational linguist. It led to agreement scores of 0.58 resp. 0.56 for the two subtasks of definition identification and classification of definition sentences as *core* or *elaborating* statements. Within the gold standard resource merged from the annotations of both raters, there where 275 definitions, 27 of which were appositive and 248 predicate-based. The average length was 1.9 sentences per definition. 208 of the predicate based definitions contained at least one core statement, 104 an elaboration, and 73 background information.

4. Definition Extraction

Based on the definition predicates and syntactic configurations that we identified in our pilot study we compiled two sets of extraction rules for the identification of definitions within decision texts. One works on the level of lemmatized and POS-tagged text, the other one presupposes dependency analyses. These are provided by the *Preds*-parser (*Preds* stands for *partially resolved dependency structure*), a semantically-oriented parsing system that has been developed in the Saarbrücken Computational Linguistics Department within the project COLLATE. It was used there for information extraction from newspaper text (Fliedner 2004). This section of the paper evaluates and compares the performance of these two extraction rule sets. The next section will look at optimizations as well as a way of acquiring further extraction rules semi-automatically.

Extractor Patterns

In order to extract sentences from a corpus by formulation patterns like the ones discussed in the previous section, these patterns have to be transformed to a format that allows using them as executable queries. For the purpose of lemma- and POS-based extraction, our corpus is stored in a MySQL-database. Search patterns are represented in a simple text-based format that allows for the specification of sets of lemma/POS-combinations together with full or partial ordering constraints that have to be present in target sentences.

A general problem of this lemma/POS-based extraction approach is the following: For most formulation types a number of different patterns have to be used due to different possible surface orders, separable verb prefixes, active/passive alternations etc. To represent the definition patterns associated with the 52 definition predicates identified in our pilot study, we therefore need the relatively large number of 93 different lemma/POS-based pattern specifications. Using dependency-parsed text as the extraction basis allows us to specify search patterns that correspond one-to-one to formulation types because surface differences are normalized in the parsing process. For the 52 definition predicates from our pilot study, we need a total of 59 search patterns (some definition predicates occur with more than one syntactic configuration).

The Preds-parser used to prepare the corpus for dependency-based extraction outputs XML-structures. Specifying Preds fragments in terms of XPath-expressions therefore offers itself as a way of formulating search templates that are executable directly (using off-the-shelf tools such as the GNU LibXML- and LibXSLT-libraries). For instance to identify definitions with the common definition predicate *vorliegen*, the XML-structures produced by the Preds parser for all sentences are first matched against an XPath-expression that selects all clause-level predicate nodes that have the stem *vorliegen*, stand in indicative present tense, have a subject and are modified by a subclause introduced by the subjunction *wenn*, or by a conjunction of such subclauses. Starting from the predicate node, the nodes corresponding to the various structural elements are then addressed by further XPath expressions.

In order to avoid unnecessary reduplication of elements that are common to different patterns, we use a condensed, XML-based pattern specification language. This language allows us to define larger numbers of patterns quickly by combining a number of separately specified common definition-bearing syntactic frames (such as the combination subject and *wenn*-subclause) and mappings (e.g. subject-to-definiendum and *wenn*-subclause-to-definiens) with lists of lemmata of potential definition predicates. The compactness and flexibility of the specification language allows for easy pattern engineering and is especially helpful for development purposes, where many different patterns have to be specified and tested quickly. Using this condensed format, the 59 patterns just mentioned are specified as 26 entries.

Evaluation

The diagram in Figure 2 compares the performance of the lemma/POS-based extractor set to that of the dependency-based extractors on the development data analyzed in our pilot study. It was obtained by sorting all extraction results according to the precision achieved by their retrieving pattern, and plotting the overall precision and recall for each top-n-segment of this ordering.

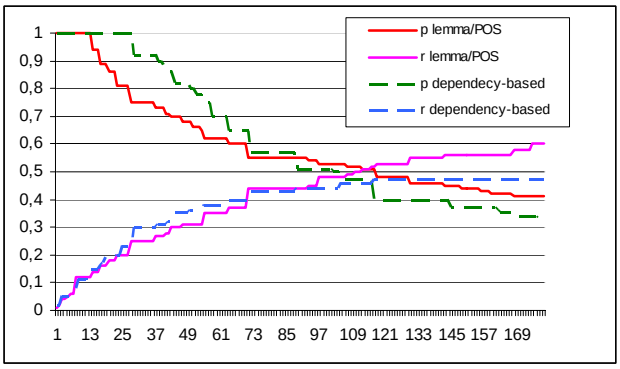

Figure 2: Performance of lemma/POS and dependency based extractors on development data

The total number of hits achieved by the lemma/POS based extractors is 1837 (thus almost half of the corpus under consideration. Only the first 180 hits are displayed in Figure 2). For this large number of hits, the recall is almost total, however the minimal precision (0.08) leads to an f-score of only 0.15. The dependency-based patterns retrieve a total of 145 hits at an f-score of 0.45. However while the precision values for the dependency-based patterns are considerably better than for the lemma/POS-based patterns for a lower number of hits, even the maximum recall reached is only less than 0.5. This problem is aggravated if we look at the performance on unseen data. Figure 3 compares both pattern sets on the goldstandard corpus.

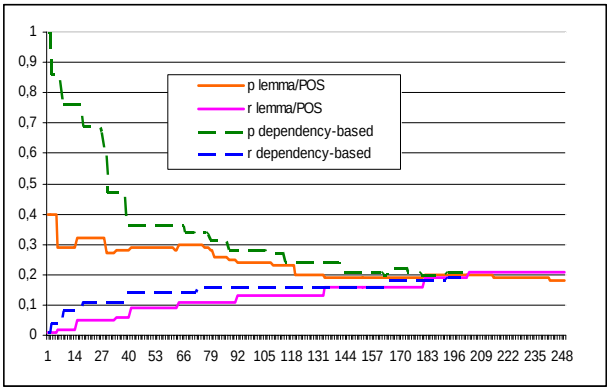

Figure 3: Performance of lemma/POS and dependency based extractors on goldstandard data

Here, the dependency-based patterns are more clearly superior to the lemma/POS-based ones. However the maximum recall reached is only 0.25, and the precision values do not reach 1 even for small numbers of hits (the f-score reached by the best subset of dependency-based patterns according to the ordering in Figure 3 is 0.24 with r=0.2 and p=0.31). The unsatisfactory recall of the dependency-based patterns on the development data is in part due to pre-processing and parser errors. Another reason is the fact that we have not used all definition predicates identified in our pilot study as extractor patterns. The deterioration of performance on the unseen corpus data points to another problematic factor: The pattern sets discussed in this section overfit the development data: On the one hand they miss out formulation types that are relevant definition indicators but do not occur in our relatively small development corpus. On the other hand some of the included patterns apparently are not generally as indicative of definitions as the development data suggested. Therefore the precision values calculated on the development corpus are not fully reliable as predictors of the actual precision that the same patterns achieve on unseen text. In the next section we are going to look at approaches that help dealing with these two problems.

5. Improvements

In this section we discuss the use of bootstrapping techniques to acquire additional extraction patterns to deal with the recall problems of our manually compiled extractor set on unseen data. We then look at a ranking scheme based on a multitude of features with automatically induced weights to improve over the prediction of precision scores from the development data.

Bootstrapping

The major problem of the manual pattern-engineering approach described in the last section is that it is hardly possible to analyse enough data manually to encounter enough of the less obvious formulations used for definitions in specific contexts. This is a general problem in text-based automatic information access. A common approach taken to deal with it is the use of bootstrapping techniques to acquire extractor patterns automatically. Bootstrapping approaches (e.g. Riloff & Jones, 1999) rely on the identification of an initial set of typical slot-fillers (*seeds*). These can then in turn be used to identify typical contexts that serve as new search patterns, which again extract further slot fillers. Various heuristics can be used to decide which fillers and patterns to actually keep and when to halt the bootstrapping cycle. Since the subparts of definitions are almost never proper names or named entities (such as the typical slot fillers in information extraction), but rather subclauses and complex NPs the main problem for bootstrapping approaches in our domain is to find a suitable granularity of seed items. Full phrases are inadequate because they do not allow for enough variability. The particular wording of a full phrase is unlikely to be shared by other definitions, and the phrase type more or less determines the syntactic frame in which it can occur. A reasonable solution is not to use full phrases as seed, but to select single seed words from these phrases according to structural criteria.

We conducted an experiment for which we selected an initial seed of 690 pairs of content words (first element from the defined phrase, second element from the defining phrase) from a collection of 138 legal definitions available online as a resource for law students.[1] For each

[1] http://www.jurawiki.de/JuristischeDefinition/AlleDefini

of these seed pairs, we then (a) scanned our full corpus (apart from the texts in the goldstandard that were to be used as test data) for parses that contain both seed words as descendants of a common governing verb node and (b) extracted as candidate patterns the respective verb nodes together with the paths to both seed words. Finally (c) we chose those 110 candidates as new extractor patterns that were as small as possible and subsumed as many of the hits from step (a) as possible. For a second run we generated new seed pairs from the extraction results retrieved by the patterns acquired in the first run by choosing all noun pairs from defined and defining phrase that co-occurred significantly more often in the extraction results than in the whole corpus. We repeated steps (a) - (c) with these new seed pairs, resulting in another 110 new extractor patterns.

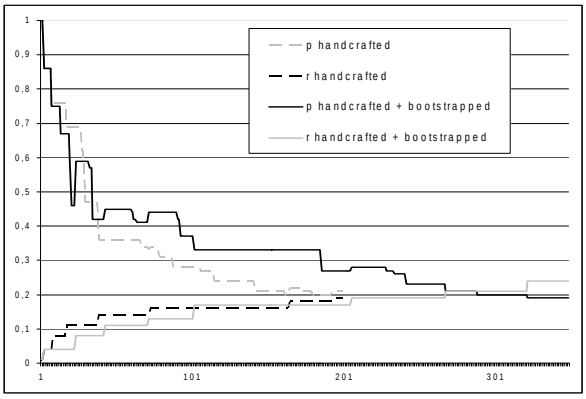

Figure 4: Performance of handcrafted and bootstrapped extractors on goldstandard data

Figure 4 compares the results of applying our handcrafted patterns on the goldstandard corpus to the hits retrieved by the bootstrapped and the handcrafted pattern set together (half of the corpus was used to estimate precisions for the bootstrapped patterns, half was used as test data. The results are averaged over two different splits). It shows that the inclusion of the bootstrapped patterns leads to a certain gain in recall (the recall of the full extractor set goes up to almost 0.5 with the bootstrapped patterns included, compared to 0.19 without them), however only if also less precise patterns are considered.

Ranking

The ranking of extraction results discussed in the previous section was constructed from precision estimates for each extractor pattern, based on its performance on the development data. As we have seen, these estimates only unreliably predict the performance on unseen data. This problem is mostly due to the small size of our development corpus and can be resolved by considering the precision of extraction results from a larger text collection. For this purpose we applied our dependency based pattern set to the full corpus (again apart from the goldstandard data) and had a graduate law student classify 3800 randomly selected hits as (non-)definitorial according to the guidelines mentioned above. From these annotated hits we calculated new precision estimates for the retrieving patterns. Additionally, we used the annotated data to train weights for a linear combination of various additional features in order to refine the ranking based on precision scores alone. These included lexical (such as the occurrence of certain stop- or boost-words, bag-of-word similarity to a set of known definitions), structural (e.g. embedding level, ordering of surface elements) and domain specific (e.g. occurrence of citations) attributes of each hit. For evaluation purposes we partitioned the 3800 annotated hits into training- and test-sets of about the same size in four different ways (randomly, but balanced over the various patterns), and ranked each of the test sets (a) according to precision estimates based on the respective training set, (b) based only on the weighted additional features, and (c) based on both sources of information. The diagram in Figure 5 compares the precision scores for all top-n segments of the rankings produced by settings (a), (b) and (c), averaged over the four training/test-splits. Setting (b) performs only slightly worse than the ordering based on precision estimates, and the combined setting (c) improves considerably over (b) as well as (a). Figure 6 shows that the combined ranking scenario is also almost consistently superior to the precision-based ranking of the extraction experiment on our goldstandard data in terms of precision as well as recall.

6. Acknowledgments

The research described within this paper is carried out within the project CORTE funded by the German Science Foundation, DFG PI 154/10-1.

7. Conclusion

In this paper we have discussed a classification of definitions in German court decisions and compared several approaches for the extraction of such definitions from large amounts of text. We have shown that it is possible to achieve a reasonable precision in this extraction task for a certain number of results using dependency-parsed text as a basis. The recall of the method can be improved by bootstrapping additional extractor patterns. Its precision can be improved using an automatically learned ranking on the grounds of combination of various additional features. It remains to be seen if an extension of the ranking scheme to bootstrapped patterns enables a better overall balance between recall and precision.

tionen

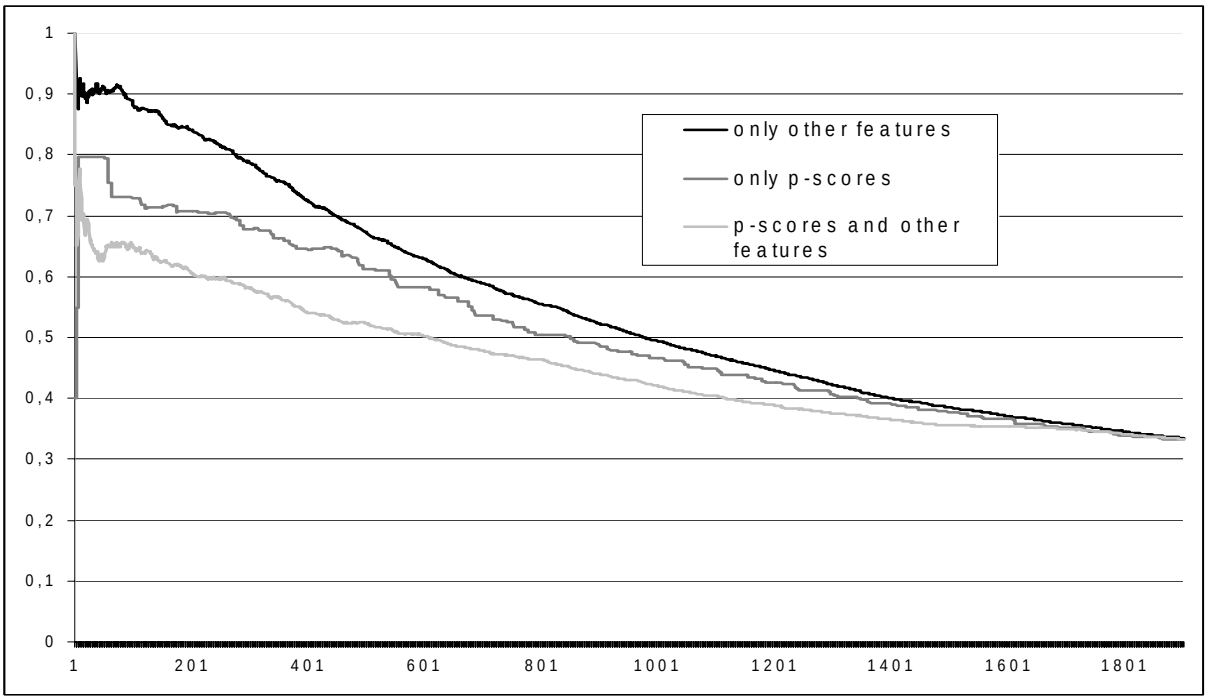

Figure 5: Precision scores with rankings on full corpus

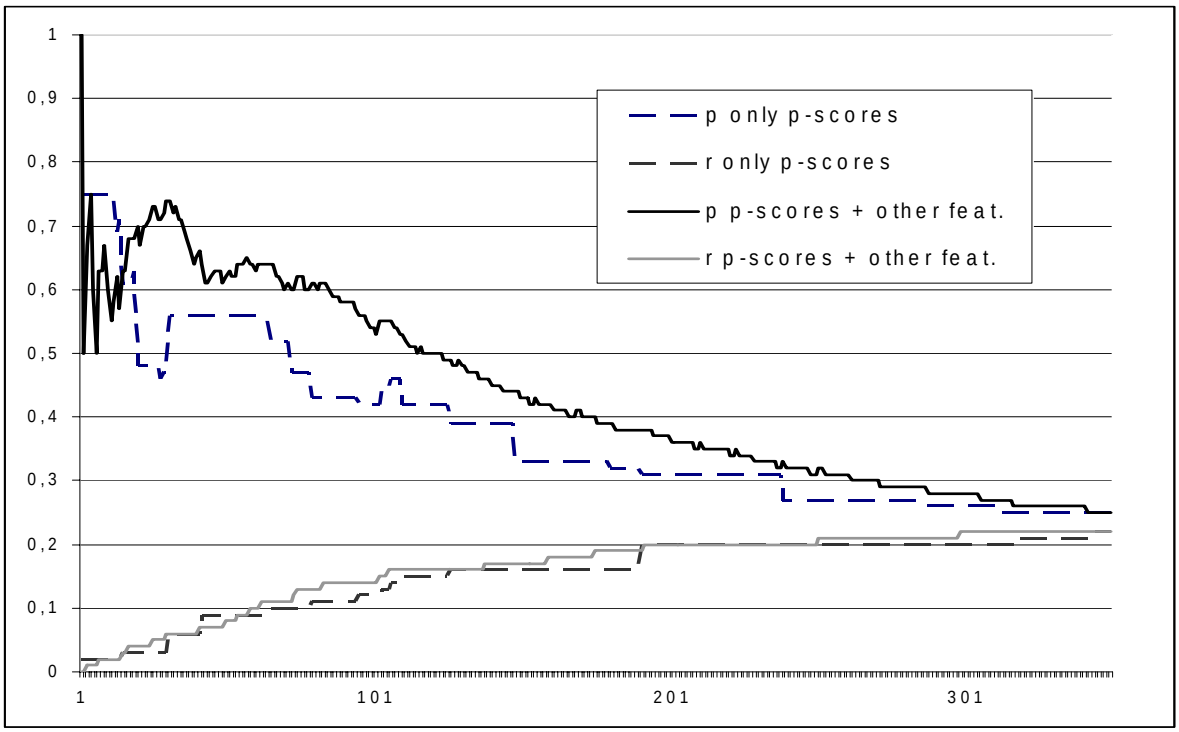

Figure 6: Precision and recall scores with ranking on goldstandard corpus

8. References

Arntz R., Picht, H. (1989). Einführung in die Terminologiearbeit. Olms

Dini, L., Liebwald, D., Mommers, L., Peters, W., Schweighofer, E. and Voermans, W. (2005). Cross-lingual legal information retrieval using a WordNet architecture. In: Proceedings of ICAIL '05: 163-167

Fahmi, I., Bouma, G. (2006). Learning to Identify Definitions using Syntactic Features. In: Proceedings of the Workshop of Learning Structured Information in Natural Language Applications, 11th Conference of the European Chapter of the Association for Computational Linguistics

Fliedner, G. (2004). Deriving FrameNet Representations: Towards Meaning-Oriented Question Answering. In Proceedings of the International Conference on Applications of Natural Language to Information Systems (NLDB). Salford, UK. LNCS 3136/2004. Springer, 64–75.

Hart, H.L.A. (1961). The concept of Law. Oxford University Press, London, UK

Hildebrandt, W., Katz, B., Lin, J. (2004). Answering definition questions using multiple knowledge sources. In: Proceedings of HLT-NAACL 2004: 49-56

Klavans, J. L. and Muresan, S. (2001). Evaluation of DEFINDER: a system to mine definitions from consumer-oriented medical text. In: Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries (Roanoke, Virginia, United States). JCDL '01. New York: ACM Press: 201-202.

Lame, G (2005). Using NLP Techniques to Identify Legal Ontology Components: Concepts and Relations. In: Lecture Notes in Computer Science, Volume 3369: 169 – 184

Landau, S. (1984). Dictionaries : the art and craft of lexicography. New York : Scribner

Lin, J., and Demner-Fushman,D. (2005). Automatically Evaluating Answers to Definition Questions. In: Proceedings of HLT/EMNLP

Meyer, I. (2001). Extracting knowledge-rich contexts for terminography. In: Bourigault, D., C. Jacquemin and M.-C. L'Homme (eds.). Recent Advances in Computational Terminology: 279–302.

Pearson, J. (1998). Terms in Context. John Benjamins, Amsterdam

Peng, F., Weischedel, R., Licuanan, A., Xu, J. (2005). Combining deep linguistics analysis and surface pattern learning: a hybrid approach to Chinese definitional question answering. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (Vancouver, British Columbia, Canada, October 06 - 08, 2005): 307-314

Riloff, E., Jones, R. (1999). Learning Dictionaries for Information Extraction Using Multi-level Boot-strapping. In: Proceedings of AAAI-99: 474 - 479

Saias, J., Quaresma, P. (2005). A Methodology to Create Legal Ontologies in a Logic Programming Information Retrieval System. In: Lecture Notes in Computer Science, Volume 3369: 185 – 200

Storrer, A., Wellinghoff, S. (2006). Automated detection and annotation of term definitions in German text corpora. In: Proceedings of LREC 2006, Genua

Valente, A. (2005). Types and Roles of Legal Ontologies. In: Lecture Notes in Computer Science, Vol. 3369: 65-76.