 ISBN 978-80-7378-250-4

Detecting Semantic Relations in Texts and Their Integration with External Data Resources

V. Kríž
Charles University in Prague, Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics, Prague, Czech Republic.

Abstract. We present our initial steps towards a linguistic processing of texts to detect semantic relations in them. Our work is an essential part of the INTLIB project whose aim is to provide a more efficient and user-friendly tool for querying textual documents other than full-text search. This tool is proposed as a general framework which can be modified and extended for particular data domains. Currently, we focus on Czech legal texts.

Introduction

Nowadays, large collections of documents form one of the main sources of information and their efficient browsing or querying is the key aspect in many areas of human activity. Existing solutions to the problem of searching large collections of documents typically implement two approaches. The *full-text search* allows the user to find documents with the highest frequency of occurrences of a specified set of keywords. The search is automatically optimized using a pre-generated index that keeps track of the occurrences of keywords. By contrast, the *metadata search* allows the user to find documents with given properties (such as, e.g., author, creation date, expiration date, list of keywords, etc.). Nevertheless, the metadata are assigned to the documents manually and, thus, inefficiently and expensively.

In general, these two approaches do not work with the semantic interpretation of the documents in the collection. For example, considering the legislation, we may need to know that the term "the High Court" means the particular institution in a particular country that has certain powers and relations to the Constitutional Court. To enable the user to access the data this way means (i) to interpret the semantics of the documents in terms of real-world objects and the relationships between them which are described in the documents, (ii) to transform the interpretation into a suitable database preferably having a standard format and standard query language, and (iii) to present the interpretation to the user in a form which enables efficient, precise and user-friendly browsing and filtering.

The main aim of the project *INTLIB — an INTelligent LIBrary* is to provide a more efficient and user-friendly tool for querying textual documents than full-text or metadata search. On the input we assume a collection of human-written documents related to a particular problem domain. INTLIB processes the data in two phases:

- *Extraction phase* — We extract a *knowledge base* from the documents. The knowledge base is a set of objects and their mutual relationships based on a particular ontology. We first exploit and utilize linguistic approaches and machine learning techniques. Then we apply algorithms for cleaning and linking the data, and their transformation to Resource Description Framework (RDF) [Beckett, 2004].
- *Presentation phase* — We deal with efficient and user-friendly visualization and browsing (querying) the extracted knowledge.

The whole system is proposed as a general framework which can be modified and extended for various data domains using plug-ins.

To depict features of the INTLIB project we use the legislation domain and we implement plug-ins that process the legislation of the Czech Republic. In spite of the fact that the Czech language processing has an extremely high prestige within NLP community [Panevová et al., 2012], NER including [Kravalova and Zabokrtsky, 2009], processing of legal documents concerns the lexicography work mainly, see [Cvrček et al., 2012] and [Pala et al., 2010].

In this paper, we focus on the initial steps we undertook in the extraction phase. We work with two types of legal documents — court decisions; acts and decreeses. (i) Court decisions consist of references to other documents. (ii) Documents are processed with a chain of linguistic tools, most of them designed and implemented as machine learning applications. We use the application `tool_chain` [Hladká et al., 2008]. Syntactic parsing is one of them [McDonald et al., 2005]. Having a syntactic parser trained on

newspapers, we are interested in its performance on legal texts. We evaluate the performance against the manual annotation of acts and decreeses.

The paper is structured as follows: In Section *JTagger* we provide a short description of the application JTagger which detects references in court decisions. In Section *Automatic Parsing and Manual Annotation of Legal Texts* we present a very short report of our research whose aim is to evaluate the performance of automatic syntactic parsing on legislative domain. In Section *Conclusion* we provide an outline of our future work.

JTagger

We approach the reference recognition as a task of Named Entity Recognition (NER) being a subtask of natural language processing, namely information extraction. The NER task detects atomic elements in text and classifies them into predefined categories such as the names of persons, organizations, locations, etc. NER systems have been designed and implemented and they use linguistic grammar-based techniques as well as statistical models (e.g. Ratinov and Roth [2009]).

Attention has been already paid to various NER tasks focusing on legal texts as evident from the paper Quaresma and Gonalves [2010] providing a comprehensive overview of projects on information extraction from the legal documents using natural language processing tools — a great majority of presented projects are on English, German, Italian and Portuguese. To our best knowledge, there is no published work addressing this issue for other Slavonic language.

A system of document reference recognition in Czech court decisions is called *JTagger*. Demo is available at `http://ufal.mff.cuni.cz/jtagger`.

Building an annotated corpus of Czech court decisions

We apply supervised learning methods thus we have to annotate data in order to train models. Currently, we focus on three types of references: (i) institutions, (ii) court decisions, and (iii) the acts published in the Collection of Laws. We also handle references/citations to specific parts of documents.

Table 1. An annotation scheme for Czech court decisions.

Tag	Description
Act	reference to **Act** or its parts
Decision	reference to court **Decision**
Effectiveness	act **Effectiveness**
Institution	**Institution**
Publisher	Institution that published a given document

We included all described entities and relations into an annotation scheme we experiment with — see Table 1 and Figure 1. According to our Annotation manual [Kríž, 2012], overlapping entities are not allowed. On the other hand, it can happen that one token can be annotated with more than one tag. For example, act reference *the Constitutional Court Act* contains the institution. The annotator marks and tags the institution first and the act afterwards. In total, we annotated a sample of 300 court decisions that are posted at the home pages of The Supreme Court (SC) and The Constitutional Court (CC). We

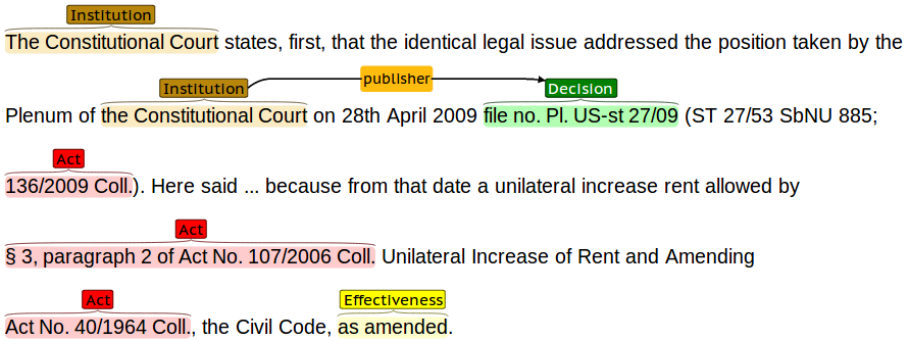

Figure 1. Annotation of court decisions.

annotated them in a web-based annotation tool *Brat* [Stenetorp et al., 2012].[1]

In order to run experiments based on the cross-validation strategy, we split the 300 annotated documents randomly into 10 folds (9 for training, 1 for testing) having quantitative characteristics presented in Table 2. Since we work with references (Act, Decision, Institution) and their attributes (Effectiveness, Publisher), we speak about entities in the next.

Table 2. Overall quantitative characteristics of training and test sets, averaged over 10 cross-validation folds.

	The Supreme Court (SC)			The Constitutional Court (CC)		
	Docs.	# of Tokens	# of Entities	Docs.	# of Tokens	# of Entities
Training set	135	332,535	8,487	135	312,191	7,910
Test set	15	36,999	943	15	34,701	879
Total	150	369,534	9,430	150	346,892	8,789

Systems

So far, we decided to compare the performance of two machine learning approaches, namely Perceptron Algorithm with Uneven Margins (PAUM) and Hidden Markov model algorithm (HMM).

PAUM. The PAUM algorithm [Li et al., 2002] is one of machine learning alternatives provided by the GATE framework[2] [Cunningham et al., 2002]. The algorithm represents a slight modification of the classical Perceptron algorithm [Kim et al., 2005] used in neural networks and extended by SVM. PAUM belongs to the category of classical propositional learners (supervised statistical classifiers) working on a set of learning features. In the GATE community, PAUM is known for providing comparable performance to SVM, with much reduced training times. In our experiments, PAUM was used in the chunk learning mode with the features listed in Table 3.

Table 3. Features used by PAUM.

PAUM model	Features
`PM small`	trigrams of word forms w_{i-2}, w_{i-1}, w_i
`PM`	5-grams of word forms $w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}$
`PM pos`	5-grams of lemmas and part of speech tags $l_{i-2}, l_{i-1}, l_i, l_{i+1}, l_{i+2}$; $t_{i-2}, t_{i-1}, t_i, t_{i+1}, t_{i+2}$
`PM pos ext`	It extends `PM pos` with an orthography feature and it distinguishes first and last tokens in a sentence.

HMM. The Czech language belongs to languages described with an attribute free word order. However, the Czech language used in legal documents in general has relatively restrictive Subject-Verb-Object word order. That is why we decided to train HMMs to recognize references.

Hidden Markov Models present historically a very first statistical model applied in the field of natural language processing [Merialdo, 1994]. In the most general case (e.g. [Jelinek, 1997]), Hidden Markov model can be described as a five-tuple (S, s_0, Y, P_S, P_Y), where $S = \{s_0, s_1, ..., s_T\}$ is the set of states, s_0 is the initial state, $Y = \{w_1, w_2, ..., w_T\}$ is the output alphabet, P_S is the set of probability distributions of transitions, and P_Y is the set of output (emission) probability distributions.

In our task, the output alphabet consists of all possible words occurring in the training data and the states contain reference tags that we assign to the words. The goal is to compute the most likely sequence of tags that has generated the input text. While PAUM models identify the beginning and end tokens for each entity, HMM annotates each token.

Experiment Evaluation and Error Analysis

We evaluate the performance of individual approaches with the standard evaluation measures which are used in information extraction and supervised machine learning.

[1] `http://brat.nlplab.org/`

[2] `http://gate.ac.uk`

When multi-token entities are evaluated, partially correct (or overlapping) matches can occur. The evaluation can be calculated in two ways, depending on what units are compared. We can use either individual (potentially multi-token) entities or tokens from which the entities are composed of. We provide the evaluation using both approaches.

Strict and *Lenient* variants of performance measures allow dealing with partially correct matches in different ways: Strict measures consider all partially correct matches as incorrect (spurious, false positive), while Lenient measures consider all partially correct matches as correct (true positive).

Cross-validation on Training Set

We wanted to investigate the statistical significance of the results, thus we performed an experiment using 10-fold cross-validation. Statistical significance was computed using the corrected resampled (two tailed) t-Test [Nadeau and Bengio, 2003], which is suitable for cross validation based experiments. Test significance threshold was 0.05.

Table 4. Cross-validation results — Strict F_1 on entities.

	Entity	HMM	PM pos ext	PM pos	PM	PM small
SC	Act	0.75±0.02	0.91±0.02 ∘	0.91±0.03 ∘	0.89±0.03 ∘	0.88±0.03 ∘
	Decision	0.82±0.08	0.97±0.02 ∘	0.96±0.02 ∘	0.95±0.03 ∘	0.94±0.02 ∘
	Effectiveness	0.89±0.04	0.90±0.05	0.89±0.05	0.88±0.08	0.82±0.10
	Institution	0.92±0.03	0.96±0.02 ∘	0.96±0.02 ∘	0.95±0.02 ∘	0.96±0.02 ∘
CC	Act	0.63±0.05	0.87±0.02 ∘	0.86±0.02 ∘	0.84±0.03 ∘	0.78±0.03 ∘
	Decision	0.83±0.05	0.95±0.03 ∘	0.95±0.03 ∘	0.93±0.03 ∘	0.92±0.03 ∘
	Effectiveness	0.96±0.03	0.96±0.03	0.96±0.03	0.96±0.03	0.96±0.03
	Institution	0.91±0.02	0.93±0.02 ∘	0.93±0.02 ∘	0.92±0.01 ∘	0.92±0.01 ∘

Error analysis

Table 4 shows the results of cross-validation, namely entity based F-measure for CC and SC decisions separately. The first column is considered as the baseline and remaining columns are evaluated against it; statistically significant increase/decrease is indicated by ∘/•, resp. We conclude that PAUM shows better performance than HMM (especially, `PM small` works with the same features as HMM and its results are better).

As a future work we want to experiment with other ML algorithms and also want to implement rule based classifier for identifying entities.

Automatic Parsing and Manual Annotation of Legal Texts

Specifying the INTLIB extraction phase, we start with a chain of sentence/token segmentation, POS tagging and syntactic parsing. For this task we used the application `tool_chain`, [Hladká et al., 2008]. However, the NLP procedures we have at our disposal are trained on newspaper texts. Since legal texts and newspaper texts essentially differ in syntactic features, we pay special attention to the verification whether we can use the parser trained on newspaper texts anyway or whether we have to do some modifications.

At least to our knowledge, very few attempts have been carried out to check the performance of parsers on legal texts. One of the main reasons is the absence of syntactically annotated gold corpora of legal texts. The first competition on dependency parsing of legal texts took place in 2012. The SPLet 2012 — First Shared Task on Dependency Parsing of Legal Texts [Dell'Orletta et al., 2012] looked at different parsing systems which have been tested on Italian and English legal data sets. However, none of the submitted systems elaborated the idea of complex sentence segmentation and modified tokenization. Instead, all of them concentrated on tuning parameters of machine learning methods they applied.

Legal texts are specialized texts operating in legal settings. In view of the fact that they should transmit legal norms to their recipients, they need to be clear, explicit and precise. However, the style of legal texts is "generally considered very difficult to read and understand".[3]

Legal texts have a very specific syntactic structure with many peculiarities. We often encounter e.g. passive voice structures, impersonal constructions, non-finite and verbless clauses, conjunctive

[3] `http://www.languageandlaw.org/LEGALTEXT.HTM`

Orig	Sample text	Compl
s_1	(1) Complex sentence:	s_1n_1
	a) first subsection,	s_1n_2
	b) second subsection,	$s_1n_3m_1$
	1. paragraph,	$s_1n_3m_2$
	2. paragraph,	$s_1n_3m_3$
	c) third subsection.	s_1n_4
s_2	(2) Simple sentence.	s_2

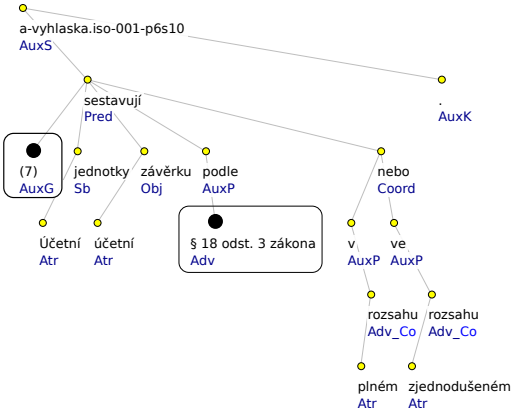

Figure 2. The example of the orginal and complex sentence segmentation and re-tokenization.

groups, such as *even though — even when — even if — as if — as though*, complex conjunctions, such as *provided (that) — granted (that), suppose (that)*, etc.).

Simple sentences in legal texts are very rare, with exception of headings, references and similar rather technical sections or their parts. Typically, the sentences are long and very complex, therefore, in order to ensure comprehensibility of the whole text they have to be clearly separated and hierarchized.

Long sentences do not necessarily obstruct the understandability of texts. Moreover, the special structure is emphasized by a significant use of punctuation such as semicolons and parentheses. Punctuation plays a crucial role because legal texts usually include very complicated syntactic patterns.

Preparing manually annotated data

We believe that manually annotated goldstandard data is necessary for precise evaluation of automatic syntactic parsers on legal texts. The process of manual annotation must be treated in a special way. The reason is that legal texts have a very specific syntactic structure leading to long dependency trees that frequently require scrolling in an annotation editor during manual annotation and their annotation is therefore a very demanding activity. To avoid this problem we propose (i) a *re-tokenization*, i.e. several tokens are joined into one token and (ii) *complex sentence segmentation*, i.e. sentences are split into smaller parts (subtrees) that are more comfortable to annotate. In the end all the subtrees are merged together in order to display the annotation of the original sentence.

We selected two legal documents from the Collection of Laws of the Czech Republic that will serve as a workbench for our study.[4] The selection was motivated in the wider context of the INTLIB project. The manual annotation is approached as a parser output checking procedure. Before parsing starts, (i) documents are processed by tokenization and sentence segmentation tuned for newspaper texts, (ii) their outputs are refined by re-tokenization and complex sentence segmentation.

Tokenization designed for newspaper texts splits all types of numbering, e.g. *(a)*, *1)* splits into *(*, *a*, *)*, *1*, *)*. Most of these tokens make the parsing harder and the annotation more confused. We propose a simple rule-based procedure that merges all originally split tokens from numbering back into one token — see the node with the form *(7)* in Figure 2.

In addition, we expressly handle references that refer either to other parts of the document or to a different document, like *§18 odst. 3 zákona*. Again, to enhance annotators' comfort, we merge such tokens into one token so we decrease the number of nodes in the dependency tree — see the node with the latter example in Figure 2.

The most important novelty in our approach is the segmentation of complex sentences into more individual parts — *segments* — because of the manual parsing that becomes more annotator friendly than the annotation of complex sentences. Figure 2 shows the differences between the original sentence segmentation and tokenization *Orig* and more advanced segmentation and re-tokenization *Compl*.

So far, we have finished the annotation of 76 *Orig* sentences that presents 6.3% of the total amount 1201 *Orig* sentences. These sentences were selected as a continuous part of the Decree on Double-entry Accounting for undertakers. They thus represent the usual mix of legal text style, from headings to

[4]The Accounting Act (563/1991 Coll., as amended) and Decree on Double-entry Accounting for undertakers (500/2002 Coll., as amended).

complex sentences describing various legal conditions and rules.

Conclusion

In this paper we present the applications, experiments, results and studies addressed in the initial phase of the INTLIB projects. We exploit and utilize linguistic approaches and machine learning techniques to obtain a knowledge base. We present the JTagger application detecting references in court decisions. Our ambition is to use the identified entities for a legal case reconstruction. We present the work on manual syntactic annotation of legal texts. We use this data to evaluate automatically assigned annotations.

Acknowledgments. I am very grateful to Jan Dědek who performed the experiments with the PAUM algorithm. I really appreciate the hard work done by Zdeňka Urešová during the annotation of legal texts. I am lucky to be a part of the INTLIB team nicely supervised by Martin Nečaský. Last but not least, I would like to thank Barbora Hladká, my PhD supervisor, for her support. The INTLIB project is funded by the Technology Agency of the Czech Republic, grant no. TA02010182.

References

Beckett, D., *RDF/XML Syntax Specification (Revised)*, W3C, `http://www.w3.org/TR/rdf-syntax-grammar/`, 2004.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V., 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)*, 2002.

Cvrček, F., Pala, K., and Rychlý, P., Legal electronic dictionary for czech, in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, edited by N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, pp. 283–287, European Language Resources Association (ELRA), Istanbul, Turkey, aCL Anthology Identifier: L12-1455, 2012.

Dell'Orletta, F., Marchi, S., Montemagni, S., Plank, B., and Venturi, G., The splet–2012 shared task on dependency parsing of legal texts, in *Proceedings of the 4th Workshop on Semantic Processing of Legal Texts 2012*, Istanbul, Turkey, 2012.

Hladká, B. V., Hajič, J., Hana, J., Hlaváčová, J., Mírovský, J., and Raab, J., The czech academic corpus 2.0 guide, *The Prague Bulletin of Mathematical Linguistics*, *89*, 41–96, 2008.

Jelinek, F., *Statistical methods for speech recognition*, MIT Press, Cambridge, MA, USA, 1997.

Kim, K., Kim, S., Joo, Y., and Oh, A.-S., Enhanced fuzzy single layer perceptron, in *Advances in Neural Networks ISNN 2005*, edited by J. Wang, X. Liao, and Z. Yi, vol. 3496 of *Lecture Notes in Computer Science*, pp. 603–608, Springer Berlin Heidelberg, 2005.

Kravalova, J. and Zabokrtsky, Z., Czech named entity corpus and svm-based recognizer, in *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pp. 194–201, Association for Computational Linguistics, Suntec, Singapore, URL `http://www.aclweb.org/anthology/W/W09/W09-3538`, 2009.

Kríž, V., *Manual for Document Reference Annotation in Czech Court Decisions*, unpublished, 2012.

Li, Y., Zaragoza, H., Herbrich, R., Shawe-Taylor, J., and Kandola, J. S., The perceptron algorithm with uneven margins, in *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pp. 379–386, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

McDonald, R., Pereira, F., Ribarov, K., and Hajič, J., Non-projective dependency parsing using spanning tree algorithms, in *Proceedings of Human Langauge Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 523–530, Association for Computational Linguistics, Association for Computational Linguistics, Vancouver, BC, Canada, 2005.

Merialdo, B., Tagging english text with a probabilistic model, *Comput. Linguist.*, *20*, 155–171, 1994.

Nadeau, C. and Bengio, Y., Inference for the generalization error, *Machine Learning*, *52*, 239–281, 2003.

Pala, K., Rychlý, P., and Šmerk, P., Automatic identification of legal terms in czech law texts, in *Semantic Processing of Legal Texts*, edited by E. Francesconi, S. Montemagni, W. Peters, and D. Tiscornia, vol. 6036 of *Lecture Notes in Computer Science*, pp. 83–94, Springer Berlin Heidelberg, 2010.

Panevová, J., Rehm, G., and Uszkoreit, H., *The Czech language in the digital age*, SpringerLink : Bücher, Springer, 2012.

Quaresma, P. and Gonalves, T., Using linguistic information and machine learning techniques to identify entities from juridical documents, in *Semantic Processing of Legal Texts*, edited by E. Francesconi, S. Montemagni, W. Peters, and D. Tiscornia, vol. 6036 of *Lecture Notes in Computer Science*, pp. 44–59, Springer Berlin Heidelberg, 2010.

Ratinov, L. and Roth, D., Design challenges and misconceptions in named entity recognition, in *Proceedings of the Thirteenth Conference on COmputational Natural Language Learning (CONLL)*, pp. 147–155, 2009.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J., brat: a web-based tool for nlp-assisted text annotation, in *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 102–107, Association for Computational Linguistics, URL `http://aclweb.org/anthology-new/E/E12/E12-2021.pdf`, 2012.