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

Vincent Kríž
Charles University in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
kriz@ufal.mff.cuni.cz
http://ufal.mff.cuni.cz/vincent-kriz

1 Introduction

In our work we focus on detecting semantic relations from unstructured texts. We present, how semantics can improve searching large collections of documents and we introduce our approach to semantic search. The essential parts of the semantic search are (1) an understanding of document semantics and (2) integration of extracting knowledge with other machine readable sources. We show, how Linked Data principles can straightforwardly do this task.

A significant amount of this work was done as a part of the INTLIB[1] project.

1.1 Motivation

In many domains, large collections of unstructured documents form main sources of information. Their efficient browsing and querying present key aspects in many areas of human activities. Typical approaches of searching large collections of documents are *full-text search* and *metadata search*. In general, both approaches do not work with the *semantic* interpretation of documents.

This disadvantage of the typical search approaches is more and more evident. According to Amerland (2013) from Google Search: „Search is changing". The way how users use search engines has changed over the past few years. Nowadays, semantic search and the ability to extract structured data from texts, represent the most accurate options for granting answers.

In addition, the wide range of devices from which users can search represents a determining factor: PCs, laptops, smartphones, tablets, TVs, etc. With the variety of devices, there are different input methods, from typing a word on the keyboard of a computer to making a request directly to voice applications.

These advances moved the search from queries like *restaurants prague*, to more specific queries, e.g. *where to eat Indian food in Prague* or *what is the best place to eat Indian food in Prague*.

Search engines have to not only identify keywords alone, but they need to understand how the data are *related*, both in the given document and through the whole collection. According to Amerland (2013), this is the most important change in the search in general – a progression from the keywords to the increasingly important entities. Words become *concepts* and search engines evolve into true learning machines.

1.2 Our Semantic Search Approach

The aim of our work is to develop approaches and systems for detecting semantic relations from texts. We see this task as one of the most important component for semantic search engines which could become more sophisticated and user-friendly for querying textual documents.

To enable users to access the *semantics* of their documents we propose (i) to interpret the semantics in terms of real-world objects and their relations, (ii) to transform this 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.

On the input we assume a collection of human-written documents related to a particular problem domain. The proposed extraction strategy consists of two phases. In the (1) *extraction phase* we extract from the documents a *knowledge base*, i.e. a set of objects and their mutual relationships, which is based on a particular ontology. In the (2) *presentation phase* we deal with efficient and user-friendly visualization and browsing (querying) of the extracted knowledge. In our work, we focused on the first, *extraction* phase.

To address the problem of suitable machine

[1] http://ufal.mff.cuni.cz/intlib

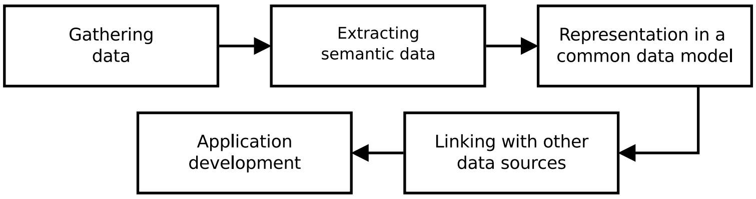

Gathering data
Extracting semantic data
Representation in a common data model
Linking with other data sources
Application development

Figure 1: A scheme of data extraction, its representation and exploitation.

readable formalism for extracted semantic data, we propose to apply Linked Data technologies. The outputs are presented in the Resource Description Framework (RDF, (Lassila and Swick, 1999)) that is, in connection with the SPARQL query language,[2] highly suitable not only as a database and querying tool, but for interpretation of the document semantics as well. Given that, RDF perfectly fits our intention to present the knowledge base according to the Linked Data principles.

Linked Open Data (LOD) is a set of principles of publishing data on the Web in a machine-readable form which enables to link related data. The links are recorded in a machine readable form and published on the Web as well as part of the data itself. LOD principles are simple:

- Use URIs to denote things.
- Use HTTP URIs so that things can be referred to and looked up by people and machines.
- Provide useful information about the things when its URI is looked up (use standards such as RDF and SPARQL).
- Include links to other related things (using their URIs) when published on the Web.

We see our work as a relationship between the fields of Information Extraction (IE) and Semantic Web (SW) as the scheme displayed in Figure 1, where the components of Gathering and Extracting data belong to IE and the components of Data representation and Data linking belong to SW. All of them are characterized by general features that are typically domain and language independent. However, their design must take into account the specification of applications that will work with the data under consideration.

1.3 Legal Domain

To depict the features of the proposed approach we use the legal domain and we implement tools that process the legislation of the Czech Republic.

We concentrate on both recognizing (1) the logical structure of legal documents which includes detecting references (links) between documents; and (2) the semantic relations between entities represented real-world objects. Both tasks should be provided automatically using textual content of given documents.

We also propose data structures which allows us to represent the recognized structures in a way suitable for further database processing.

From the Linked Open Data point of view, *things* mentioned in Linked Data principles are legal documents and their parts. Links between the things are relations (e.g., a section is *a part of* an act, an act *amends* another act, a court decision *cites* a section of an act, a court decision *cancels* another court decision, etc.). Applying the principles to the legal domain therefore means assigning HTTP URIs to legal documents and their parts, representing their metadata in RDF, extracting relationships among the documents from their original textual content and publishing all data so that the documents and their parts can be accessed via dereferencing their HTTP URIs or using the SPARQL query language.

1.4 Thesis Structure

Since the development and implementation of the proposed semantic search system presents a huge amount of work and it is mostly technical, we split the task into several sub-tasks. In this thesis we focus on the two most interesting sub-tasks from the point of Natural Language Processing (NLP).

In Section 2 we describe the sub-task of detecting references in court decisions. The system outputs are used to enrich a logical structure of legal documents. Section 3 presents a sub-task of detecting semantic relations in legal documents. The

[2] http://www.w3.org/TR/rdf-sparql-query/

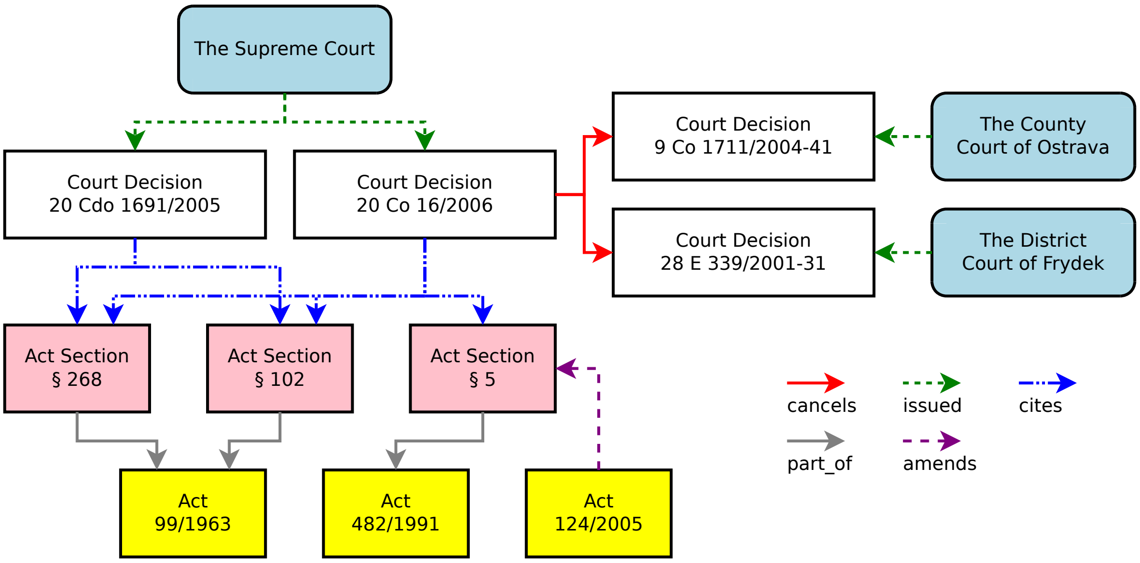

The Supreme Court
Court Decision 20 Cdo 1691/2005
Court Decision 20 Co 16/2006
Court Decision 9 Co 1711/2004-41
Court Decision 28 E 339/2001-31
The County Court of Ostrava
The District Court of Frydek
Act Section § 268
Act Section § 102
Act Section § 5
Act 99/1963
Act 482/1991
Act 124/2005
cancels
issued
cites
part_of
amends

Figure 2: Sample Links in Czech court decisions.

proposed system uses queries over dependency trees. Since dependency parsing has the most influence on the performance, in Section 4 we address the issue of the parsing of Czech legal texts. In Section 5 we list our plans for improving proposed methods.

2 Detecting References in Court Decisions

In this section we describe our work on the task of detection and classification of references in Czech court decisions. We present how structured data and their integration could help users in their work with court decisions and how to obtain more precise and relevant results of their search over this domain.

We present a statistical system *JTagger* for detecting entities in court decisions. We approach the task using machine learning methods and report F-measure over 90% for each entity. The results significantly outperform the systems published previously. More details about our system were published by Kríž et al. (2014a).

2.1 Motivation

Our aim is to increase one's comfort when searching in collections of court decisions. Because of the complexity of various legal documents, it is very hard for users, i.e. legal professionals as well as common citizens, to search for the required documents.

We demonstrate the most common use cases of how professionals as well as citizens work with court decisions and documents published in the Collection of Laws of the Czech Republic.

Acts, decrees and other documents from the Collection of Laws of the Czech Republic are usually structured to sections which may contain further subsections. In addition, a document may contain references to other documents. A reference may target not only a whole document but also its particular section. Therefore, the structure encoded in documents and references between them form a complex network (see Figure 2 for an example). Moreover, related documents are often published by different public authorities.

It would be useful to enable to browse this distributed network among different data sources and search for relationships between documents and/or their parts. Examples of common use cases are presented in the following list. For the demonstration, we use Figure 2, firstly published by Nečaský et al. (2013). It shows a part of the network comprising several related acts and court decisions.

1. A user is reading a particular section of an act (e.g., *Act Section §102* of *Act 99/1963*). He would like to see what court decisions have been made in the last decade related to this particular section (i.e., decisions *20 Cdo 1691/2005* and *20 Co 16/2006*).

2. A user is reading a particular section of an act (e.g., *Act Section §5* of *Act 482/1991*). He would like to find out what amendments cor-

recting the act the chosen section belongs to came to force in 2005 (i.e., an amendment of *Act 482/1991* defined by *Act 124/2005*).

3. A user received a court decision (e.g., *20 Cdo 389/2004*). There are various references to other court decisions and also sections of acts and amendments encoded in the text. He would like to see the reading of each of the referenced decisions and sections.

All these use cases are problematic because documents are published as unstructured textual documents by various authorities at different places of the Web. Moreover, the sources are not interlinked at all. Their logical structure (sections and their subsections) and links between them are encoded in the text in a way which can only be interpreted by a human. Therefore, the user can only read the sources and has to search for relationships manually. This is very time consuming, cumbersome and the user omits important relationships very easily.

We see the task of detection and classification of references in legal documents as the first step towards a building of the network described in the previous section. In our work we focus on the Czech court decisions and we propose a system for detecting references to other court decisions and acts.

The text of decisions is typically more verbose, less formally structured and the details of a reference are often mixed with a sparse text or expressions leaving important details as implicit, causing ambiguity not easy to solve.

In the system of courts of the Czech Republic, only decisions of the Constitutional and the Supreme Court are available on-line.[3] None of them has a unified style of citations. Even more, there are different opinions what to cite. Some judges cite other court decisions only, some of them cite various types of the literature as well, some of them cite *everything* (blogs, internet sources, Bible, novels, etc.).

2.2 Related Works

We published a comprehensive overview of related work in (Kríž et al., 2014a).

In the past decade, several approaches to the entity recognition in legal texts were reported. Dozier et al. (2010) distinguish three methods – lookup, rule-based and statistical models. In addition, the methods can be combined in a number of ways.

The *lookup method* creates a list of entities and then simply tags all mentions of entities in texts. However, law names (or names of any other regulation) do not follow a unique pattern. They can even contain commas and other names, which make the entity detection task more difficult. In addition, the lookup method may generate many false positives if a list of entities contains many ambiguous words. Applying this method on flective languages requires manipulating several word forms per lemma and the lemmatization makes this method language dependent. Another drawback of this approach is that if a name is not in the list, it will not be recognized. In addition, within a document, new law names may be defined (typically abbreviations and acronyms). These names will be missed unless they are added to the list.

By looking at the development data, one can define a *rule-based* system with a set of rules that recognize the majority of entities in the data and do not produce many false positives. Development of rule-based systems requires manually annotated development data and a large amount of effort from experienced rule writers. Even more, maintenance of such rule sets can be tricky because rules often intricate interdependencies that are easy to forget and make modification risky.

Statistical models offer an alternative to contextual rules for encoding contextual cues. One way of thinking about such statistical models is as a set of cues that receive weights and these weights are combined based on the probability and statistical concepts. A knowledge engineer must develop features that correspond to cues, pick the appropriate statistical model, and train the model using training data. Development of statistical models requires manually annotated training data and a large amount of effort from an experienced machine learning expert. Adding new development data is definitely more straightforward than editing contextual rules.

Table 1 presents relevant systems for detecting references in legal English, Italian and Dutch texts developed recently. The systems apply different detecting techniques, like lists, POS tagger, parser, regular expressions and they belong to either hybrid or rule-based strategies. Their evaluation was performed on different data sets. We provide re-

[3] `http://usoud.cz`, `http://nsoud.cz/`

System	Lang.	Tools	Technique	Acc.	Prec.	F-1
(Dozier et al., 2010)	ENG	Lists	Hybrid			85 %
(Bruckschen et al., 2010)	ENG	POS tagger	Rule-based			34 %
(Quaresma and Gonçalves, 2010)	ENG	Parser	Rule-based		35 %	
(Bacci et al., 2012)	ITA	Regexps	Rule-based			
(Palmirani et al., 2003)	ITA	Regexps	Rule-based			85 %
(DE et al., 2006)	DUT	Regexps, Lists	Rule-based	95 %		

Table 1: An overview of systems for the reference detection task in legal texts. Their evaluation was performed on different data sets. We provide reported accuracy (Acc.), precision (Prec.) and F-measure (F-1).

ported accuracy (Acc.), precision (Prec.) and F-measure (F-1).

2.3 System

Data To obtain the training and the test set for the experiments, we manually annotated the sample documents. We prepared a sample of 300 court decisions published on-line by the Supreme Court and the Constitutional Court. The annotated documents are available on-line.[4] The sample of the annotation is presented in Figure 3.

Models We experimented with the tagger based on Hidden Markov Models (HMM). HMMs present historically a very first statistical model applied in the field of natural language processing (Merialdo, 1994). 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.

In the INTLIB project, we experimented with several other models and learning algorithms. The most successful model uses Perceptron Algorithm with Uneven Margins (PAUM). The most important difference between systems is, that PAUM identifies the beginning and end tokens for each entity, but HMM annotates each token.

Evaluation and Error Analysis We evaluate the performance of individual approaches using the 10-fold cross-validation and standard evaluation measures. Table 2 shows the F-measure for the Constitutional Court (CC) and the Supreme Court (SC) decisions separately. The results are presented in a form of confidence intervals. The first column (HMM) is always the baseline and remaining columns are evaluated against it; a statistically significant increase/decrease is indicated by ○/●, resp.

We can formulate a conclusion that PAUM shows better performance than HMM (especially, PM small works with the same features as HMM and its results are better).

We manually checked the output of algorithms and we identified the following rather frequent errors: (i) References labeled with two separate tags instead of one tag. For example, in the reference *file no. 7 To 346/2011*, the token *To* is not recognized as a part of a document reference. (ii) An institution's name ends with a number, like *Disctrict Court for Prague 4*, and the last token *4* is not recognized as a part of the reference entity. (iii) Names of foreign courts, e.g. *Land Court in Norimberg, Germany*.

At least to our knowledge, there exists no other system for the reference detection in legal texts employing statistical models. We achieved performance that outperforms all results published in literature so far.

The demo, data and source codes are available at

`http://ufal.mff.cuni.cz/jtagger.`

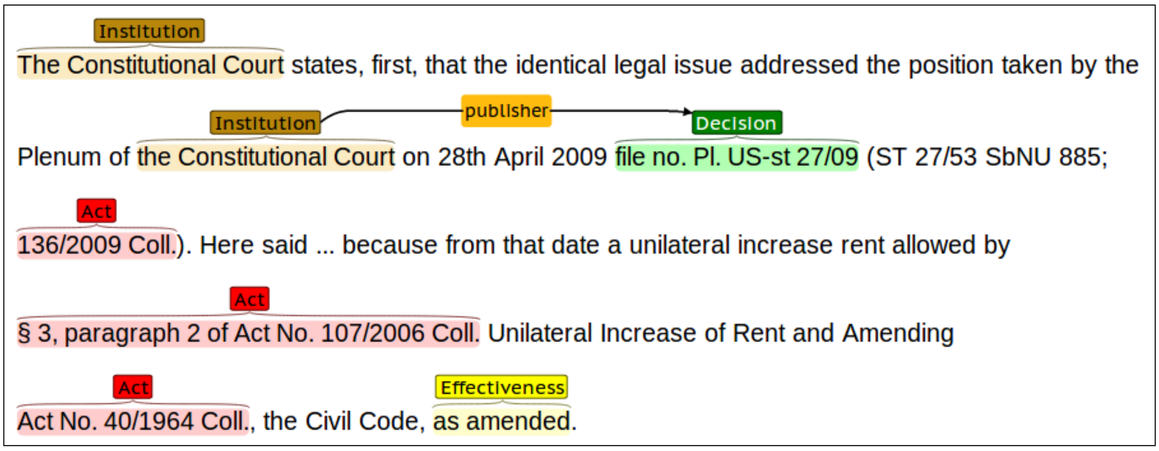
2.4 System Integration

The JTagger system was proposed and implemented as a component of a pipeline for processing legal documents. The rest of the pipeline was implemented as a part of the INTLIB project. The output of the pipeline is a logical network between legal documents. The network is formally defined by ontology proposed by Nečaský et al. (2013).

Technically, the pipeline is defined in the OD-CleanStore system.[5] This system publishes court decisions according to the principles of Linked Data. Every day, new decisions published by the Supreme Court and the Constitutional Court are

[4] `http://ufal.mff.cuni.cz/jtagger`

[5] `http://sourceforge.net/projects/odcleanstore/`



The Constitutional Court states, first, that the identical legal issue addressed the position taken by the Plenum of the Constitutional Court on 28th April 2009 file no. Pl. US-st 27/09 (ST 27/53 SbNU 885; 136/2009 Coll.). Here said ... because from that date a unilateral increase rent allowed by § 3, paragraph 2 of Act No. 107/2006 Coll. Unilateral Increase of Rent and Amending Act No. 40/1964 Coll., the Civil Code, as amended.

Figure 3: An annotation of court decisions.

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 ○

○, ● statistically significant improvement or degradation w.r.t. HMM

Table 2: The cross-validation results of the most successful models.

automatically processed by JTagger and converted to RDF.

3 Extracting Knowledge from Unstructured Texts

The system JTagger proposed in Section 2 helped us to build a network of legal documents. Besides the logical structure and links, legal documents contain also semantic information.

In this section we present a system RExtractor that extracts a *knowledge base* from raw unstructured texts. The knowledge base is a set of entities and their relations and represented in an ontological framework. The RExtractor system implements an extraction pipeline. The pipeline processes input texts by linguistically-aware tools and extracts entities and relations using queries over dependency trees. The system is designed both domain and language independent, however we demonstrate it on processing Czech legal texts.

The work presented in this section was published by Kríž et al. (2014b). In addition, the system was accepted to the system demonstrations session at the 2015 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL HLT 2015).

3.1 Motivation

Acts and other legal documents define rights and obligations of natural and legal persons. Different documents define different rights and obligations for the same kind of natural or legal person or for different persons which are, however, semantically related (e.g. one person is a special type of another person and it *inherits* the rights and obligations). Therefore, the rights and obligations of persons defined by acts and other legal documents form a complex network, similar to the described network of links among legal documents. In this case the network is defined by the semantic information encoded in the documents and we can therefore speak about a semantic network or a knowledge graph. Again, it would be useful for users to be able to browse and query such network. Holubová et al. (2014) list some sample common use cases and demonstrate them in Figure 4:

- A user wants to know what are the obliga-

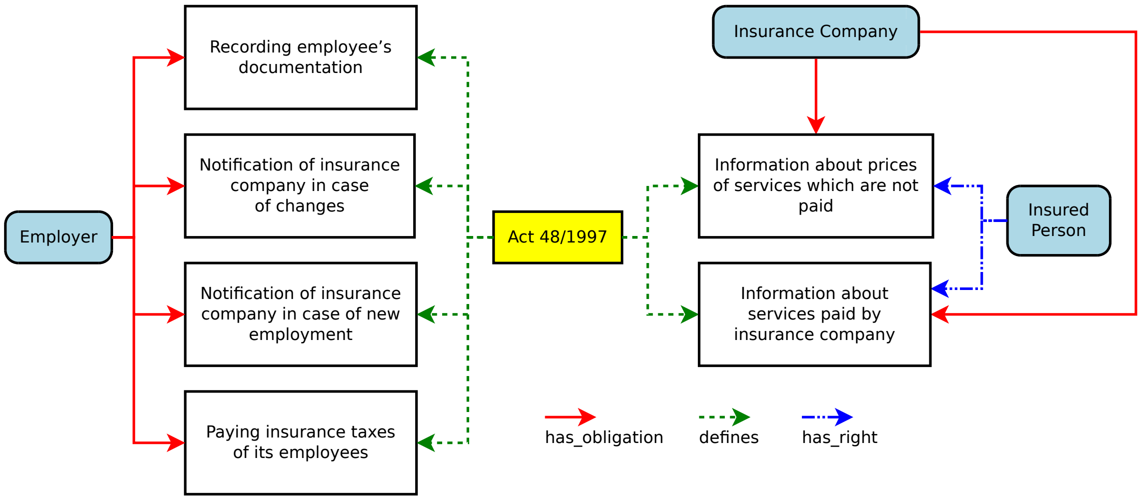

Employer
Recording employee's documentation
Notification of insurance company in case of changes
Notification of insurance company in case of new employment
Paying insurance taxes of its employees
Act 48/1997
Insurance Company
Information about prices of services which are not paid
Information about services paid by insurance company
Insured Person
has_obligation
defines
has_right

Figure 4: A sample of semantic concepts extracted from the Public Health Act valid in the Czech Republic.

tions of his employer regarding his health insurance. For example, according to the sample network depicted in Figure 4, the user can get information that his employer has an obligation to record employees documentation, notify insurance company about changes in case of changes in employees information, etc.

- A user wants to know what kind of information his health assurance company has to provide him. For example, according to Figure 4, the user can see that he has the right to obtain information from his insurance company about services provided and paid by the company as well as information about prices of services which are paid by him.

Nečaský et al. (2013) proposed the ontology for representing the structure of Czech legal documents. Our motivation is to enrich this ontology with semantic information to provide users with more intelligent search in documents.

We demonstrate the system for the legislative domain, namely we concentrate on acts, decrees and regulations published in the Collection of Laws of the Czech Republic. Although there are several systems where users can browse Czech legal texts (e.g. ASPI[6] or ZákonyProLidi.cz[7]), the systems do not offer any additional information, for example hyperlinks to referred documents.

3.2 Related work

We provided a detailed research of related work in (Kríž et al., 2014b).

The extraction of relational facts from raw texts has been of interest in information extraction for the last decade. With the emergence of the Semantic Web (Berners-Lee et al., 2001) and ontologies (Biemann, 2005), data integration has become an additional challenge. There has been a considerable amount of research on applying semi-supervised methods for data integration (Carlson et al., 2010). Unsupervised approaches have contributed further improvements by not requiring hand-labeled data (Fader et al., 2011).

Chiarcos et al. (2012) document the recent applications of Linked Data in NLP. Authors focus on language archives for language documentation, typological databases, lexical-semantic resources in NLP, multi-layer annotations and semantic annotation of corpora.

An elaborated overview of current efforts in a legal text processing is given in (Francesconi et al., 2010). The main issues include information extraction, construction of knowledge resources, automatic summarization and translation. The processing of Czech legal texts has been overviewed during the work on the Dictionary of law terms (Pala et al., 2010). Processing of non-Czech legal texts is established as well, see e.g. (Francesconi et al., 2010) for a review of current efforts.

[6] http://systemaspi.cz
[7] http://zakonyprolidi.cz

3.3 System

We have proposed a general, domain and language independent architecture. In this section we provide both (1) a general description of implemented methods, and (2) a description of adaptations done for Czech legal documents. The system architecture is displayed in Figure 5 and it consists of four components:

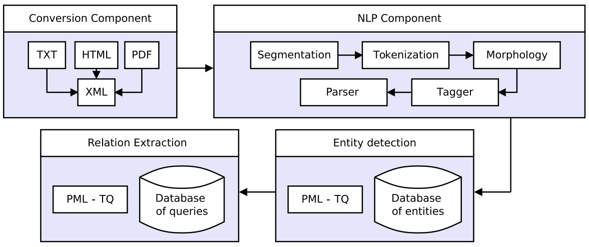


Figure 5: RExtractor architecture.

Conversion A largely technical component converting various input formats into internal representation. Although legal texts under consideration have strictly hierarchical structure, there is no official machine readable source of them with their structure. Therefore, we converted the input texts according to the RExtractor XML Schema.

Natural Language Processing A linguistic component providing various analyses of input texts, namely sentence segmentation, tokenization, morphological analysis, part-of-speech tagging, and dependency parsing. The employed procedures fit the framework originally formulated in the Prague Dependency Treebank (Hajič et al., 2006; Bejček et al., 2013).[8] The procedures are available in the Treex framework (Popel and Žabokrtský, 2010).

We can see the dependency parsing as a key procedure employed in RExtractor. NLP procedures we have at our disposal for Czech are trained on newspaper texts.[9] We pay a special attention to the verification whether we can use the parser trained on newspaper texts or some modifications are needed. We address this issue in detail in Section 4 of this work.

Entity Detection An extraction component querying dependency trees to detect entities stored in Database of Entities (DBE, see Figure 5) in texts and it exploits the PML-TQ tool (Pajas and Štěpánek, 2009).[10] DBE is built by domain experts.

We asked experts from the accounting domain to manually annotate one document[11] from the Collection of Laws of the Czech Republic. Manually recognized entities in the decree were automatically parsed to import their dependency trees into DBE. Consequently, the queries for entity detection were automatically generated from these trees. In (Kríž et al., 2014b), we provide evaluation and error analysis of the proposed entity detection.

Relation Extraction An extraction component querying dependency trees with highlighted entities to detect relations between them. It exploits the PML-TQ tool as well and poses queries stored in Database of Queries (DBQ, see Figure 5). DBQ is built by both domain and PML-TQ experts.

For the legal domain, we focus on three different types of relations: *definitions* (D) – sentences where entities are explained or defined; *Obligations* (O) – sentences bearing the information *Entity* is obligated to do *Something*; *Rights* (R) – sentences bearing the information *Entity* has a right to do *Something*. Tree queries for detecting these relations are designed manually by both domain and PML-TQ experts and respect the strategy to cover the maximum number of relations with the minimum number of queries.

Illustration Let's assume this situation. A domain expert is browsing a law collection. He is interested in the *to create something* responsibility of any body. In other words, he wants to learn *who creates what* as is specified in the collection. We illustrate the RExtractor approach for extracting such information using the sentence *Accounting units create fixed items and reserves according to special legal regulations*.

Firstly, the NLP component generates a dependency tree of the sentence, see Figure 6. Secondly, the Entity Detection component detects the entities from DBE in the tree: *accounting unit*, *fixed item*, *reserve*, *special legal regulation* (see the highlighted subtrees in Figure 6). Then an NLP expert formulates a tree query matching the domain expert's issue *who creates what*. See the

[8] http://ufal.mff.cuni.cz/pdt3.0/

[9] http://ufal.mff.cuni.cz/pdt2.0/doc/pdt-guide/en/html/ch03.html

[10] http://ufal.mff.cuni.cz/pmltq/

[11] The Decree on Double-entry Accounting for undertakers (500/2002 Coll., as amended).

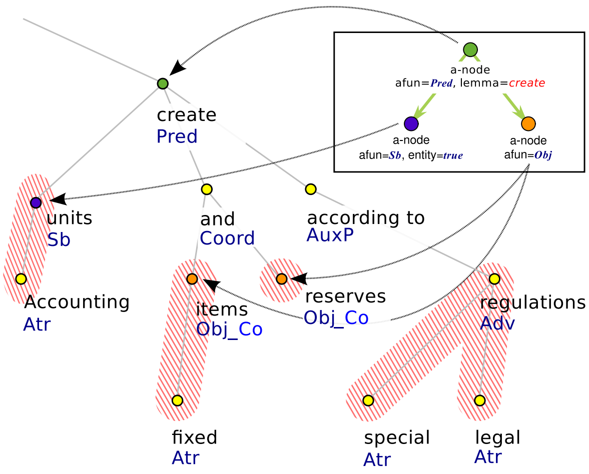


Figure 6: The extraction of *who creates what*.

Subject	Predicate	Object
accounting unit	create	fixed item
accounting unit	create	reserve

Table 3: Data extracted by the query displayed in Figure 6.

query at the top-right corner of Figure 6: (1) he is searching for *creates*, i.e. for the predicate having lemma *create* (see the root node), (2) he is searching for *who*, i.e. the subject (see the left son of the root and its syntactic function afun=Sb), and *what*, i.e. the object (see the right son of the root and its syntactic function afun=Obj). Even more, he restricts the subjects to those that are pre-specified in DBE (see the left son of the root and its restriction entity=true). Finally, the Relation Extraction component matches the query with the sentence and outputs the data presented in Table 3.

A domain expert could be interested in more general responsibility, namely he wants to learn *who should do what* where *who* is an entity in DBE. A tree query matching this issue is displayed in Figure 7. The query is designed to extract (*subject*, *predicate*, *object*) relations where the *subject* is the object in a sentence. We extract the data listed in Table 4 using this query for entity-relation extraction from the sentence *The proposal for entry into the register shall be submitted by the operator*.

Evaluation We used one legal document[12] for the manual query development. We carried out the evaluation on the another document[13] where we manually detected relations. The system achieved precision of 80% and recall of 63%. Our preliminary results are in line with already published work, e.g. (Exner and Nugues, 2012). Almost all related systems report recision higher than recall.

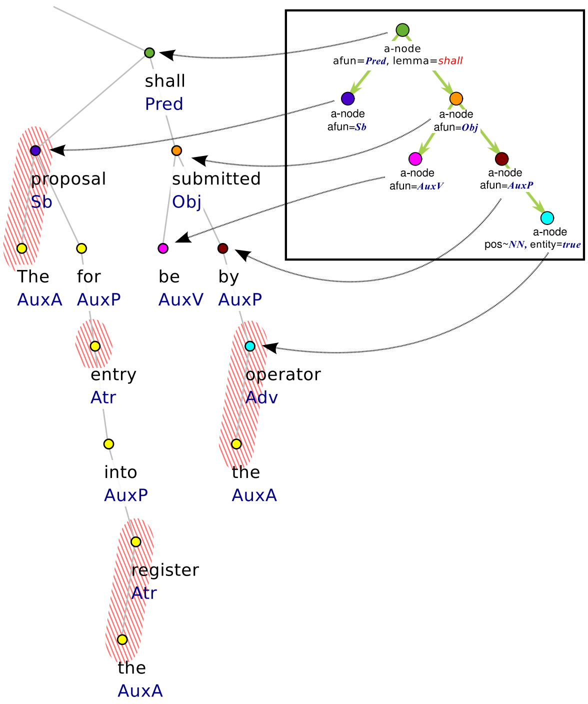


Figure 7: The extraction of *who should do what*.

Subject	Predicate	Object
operator	submit	proposal

Table 4: Data extracted by the query displayed in Figure 7.

From the error analysis, we can conclude, that missing relations are caused (1) by errors in syntactic parsing and (2) by the tree query desing, which do not try to cover all, but just most frequent syntactic patterns.

We can see the results as very promising because, (1) syntactic parsers could be adapted more on legal domain, and (2) missing queries could be defined when we relax the strategy to cover just most frequent syntactic patterns.

3.4 Integration

We used the outputs of the system proposed in this section and enriched the ontology proposed by Nečaský et al. (2013). The new version of the

[12] The Accounting Act (563/1991 Coll., as amended)

[13] The Decree on Double-entry Accounting for undertakers (500/2002 Coll., as amended).

ontology represents logical structure of acts and consolidated expressions as well as new semantic relations.

The extension has two parts. We describe each as a separate ontology: (1) *Legal Concepts Ontology* with URI http://purl.org/lex/ontology/concepts#. The ontology enables to represent the extracted entities and relationships between them independently of the original text of the ontology. (2) *Lingvistic Ontology* with URI http://purl.org/lingv/ontology#. The lingv: ontology enables us to display text chunks from which RExtractor extracted particular concepts and relations.

4 Dependency Parsing of Czech Legal Texts

In Section 3 we describe the RExtractor system. Now, we pay a special attention to the automatic dependency parsing as it has the most influence on the system performance.

We exploit sources that are already available, namely a corpus-based parser trained on Czech newspaper texts (McDonald et al., 2005). Since legal texts and newspaper texts essentially differ in syntactic features, we pay special attention to the examination whether we can use the parser trained on newspaper texts or whether we have to do some modifications.

4.1 Introduction

This issue falls into the task of domain adaptation where one major approach to improving parsing accuracy is to provide better model for certain domains. The idea is that a parsing model is trained on one type of text and must be applied to a text from a different domain which contains syntactic, stylistic, and lexical changes that are to be adjusted by models (see, for example (McClosky et al., 2010)). Currently, we concentrate on evaluating the application of a parser for one domain to another, not on analysing the models.

Dependency parsing of Czech legal texts fits the framework originally formulated in the Prague Dependency Treebank (PDT) project.[14] The dependency approach to syntactic analysis with the main role of the verb is applied. Technically, we speak about the *analytical* (*a*-) layer of annotation[15] where each token in the sentence has one corresponding node and dependencies are assigned with the syntactic dependency function stored in the afun attribute.

We carry out the examination in three steps: (1) we apply the already existing deep syntactic parser (McDonald et al., 2005) on a sample of legal texts, (2) we manually check and correct the parser output, (3) we quantify the parser performance against the manual annotation.

4.2 Syntax of the Czech Legal Texts

Legal texts are specialized texts operating in legal settings. They should transmit legal norms to their recipients, therefore they need to be clear, explicit and precise. However, the style of legal texts is ”generally considered very difficult to read and understand”.[16]

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 and conjunctive groups. Simple sentences are very rare. Typically, the sentences are long and very complex. Punctuation plays a crucial role because legal texts usually include very complicated syntactic patterns. The complexity of sentences found in legal texts is exemplified in Table 5 which shows a sentence from the Collection of Laws of the Czech Republic.[17]

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 against 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.

[14] http://ufal.mff.cuni.cz/pdt3.0

[15] http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/a-layer/html/index.html

[16] http://www.languageandlaw.org/LEGALTEXT.HTM

[17] Act on Customs Administration in the Czech Republic (§ 4, 17/2012 Coll., as amended).

(1) Generální ředitelství cel a) **vykonává** působnost správního orgánu nejblíže nadřízeného celním úřadům, b) **převádí** cla podle přímo použitelného předpisu Evropské unie, c) **stanovuje**, ve kterých věcech v oboru působnosti orgánů celní správy jde o případy celostátního nebo mezinárodního významu, d) **je** orgánem celní správy, který má ve věcech vymezených trestním řádem postavení policejního orgánu (dále jen pověřený celní orgán), jde-li o případy celostátního nebo mezinárodního významu, e) **plní** funkci centrální analytické jednotky pro účely analýzy rizik.
(1) The General Directorate of Customs a) **is** an administrative body exercising superior authority to customs offices, b) **administers** the customs duty in compliance with the relevant EU regulation, c) **determines** which cases under the remit of customs authorities are of nationwide or international importance, d) **is** a customs authority with the competences of a police authority (hereinafter referred to as competent customs authority) as defined in the penal code when dealing with cases of nationwide or international importance, e) **functions** as a central analytical body analyzing risks.

Table 5: An example sentence from the Collection of Laws of the Czech Republic

4.3 Manual Annotation

We selected two legal documents from the Collection of Laws of the Czech Republic that serve as a workbench for our study.[18] The selection was given by the goals determined in the INTLIB project, namely focusing on the accounting subdomain.

Figure 8 visualizes the steps we undertook. Before the parsing starts, tokenization and sentence segmentation tuned for newspaper texts are applied. Then their outputs are refined to meet special features of legal texts – see 'complex sentence segmentation' and 're-tokenization'. Since we want to examine how to parse legal texts effectively using a parser trained on texts from a different domain, we need to have a gold standard annotation. We got it by checking the parser output.

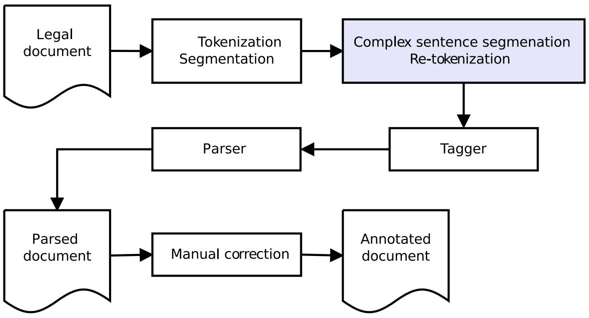


Figure 8: A scheme for legal text processing

With respect to the discussion on the complexity of legal text sentences in Section 4.2, we can see two main reasons why both sentences and tokens before they are processed either by parser or annotator need to be treated specially: (i) awareness of typical errors the parser produces; (ii) annotators' comfort.

Complex sentence segmentation We propose automatic procedure which split complex sentences into more individual parts, so called *segments*. They might not be complete sentences nor even complete clauses. The manual annotation of segments becomes more annotator friendly than the annotation of complex sentences. Table 6 shows differences between the original sentence segmentation $Orig$ and more advanced segmentation $Compl$.

We finished the annotation of 1,133 $Orig$ sentences. Out of them, 101 complex sentences were identified and segmented into 536 segments. Therefore we work with 1032 $Orig$ (non-segmented) sentences and 536 segments. Table 7 contains the number of segments into which the complex sentences were split. In addition, this table shows how many sentences were segmented into a particular number of segments.

Re-tokenization We designed re-tokenization as joining of tokens. Doing it, we decrease the number of nodes in dependency trees, i.e. we increase the annotator comfort.

Tokenization designed for newspaper texts splits all types of numbering, e.g. it splits *(a)*, *1)* into *(*, *a*, *)*, *1*, *)*. Most of these tokens make the parsing harder and the annotation more con-

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

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

Table 6: Orginal vs. complex sentence segmentation

n **# of segments**	**# of sentences with n segments**
24	1
14	1
13	1
12	1
11	4
10	1
9	7
8	3
7	5
6	6
5	12
4	28
3	27
2	4
1	1032

Table 7: Segmented sentences

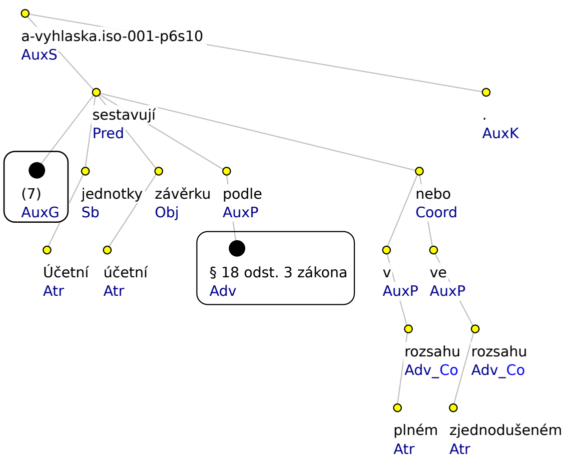


Figure 9: Re-tokenization

# of tokens	**# of *Orig* sentences**	**(# of non-segm. sentences) + (# of segments)**
1-10	570	509
11-20	446	418
21-30	391	330
31-40	165	157
41-50	86	78
51-60	42	40
61-70	16	20
71-80	13	8
$\geq$ 90	6	8

Table 8: Sentences and segments of a given length

fused. We propose a simple rule-based procedure that merges all originally split tokens from numberings back into one token - see the node with the form *(7)* in Figure 9. We manipulate references that refer either to other parts of the document or to a different document in the same way as numberings – see the node with the form §*18 odst. 3 zákona* in Figure 9. Changes in the length of sentences and segments are listed in Table 8.

In sum, 536 segments and 1032 non-segmented sentences contain 35,085 nodes. Over one third of the nodes (9,198) comes from segments, while the remaining 25,887 have been in the non-segmented sentences. Average sentence length for the non-segmented ones was 25 tokens, while each of the long, segmented sentences contained 91 tokens (17 per segment). The most segmented sentence has been split into 24 segments with 491 nodes, and the least segmented one had 2 segments. The largest segment had 142 nodes.

All the segments and non-segmented sentences were processed by the parser. We carefully tracked all the changes the annotator made while checking the parser output. We evaluated both the changes in the syntactic dependency function assignment as well as head-dependent relation changes (dependency errors/changes).

In addition to the tree correction, the annotator added inter-segmental links for nodes the head of which is in a different segment. Figure 10 displays dependency trees of segments belonging to the complex sentence presented in Table 5. For simplicity, we consider only 4 out of 6 segments (*(1), a), b), e)*). According to the PDT annotation guidelines,[19] this sentence should be annotated as a coordination of predicates (i.e. *is*, *administers*, *functions*) where the comma in the segment s_1n_3 is its head.

We use references set by the annotator to link nodes with their proper parent nodes if they are in different trees representing the segments. In Figure 10, references are represented by blue arrows pointing from children to their parent nodes.

[19] `http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/a-layer/html/index.html`

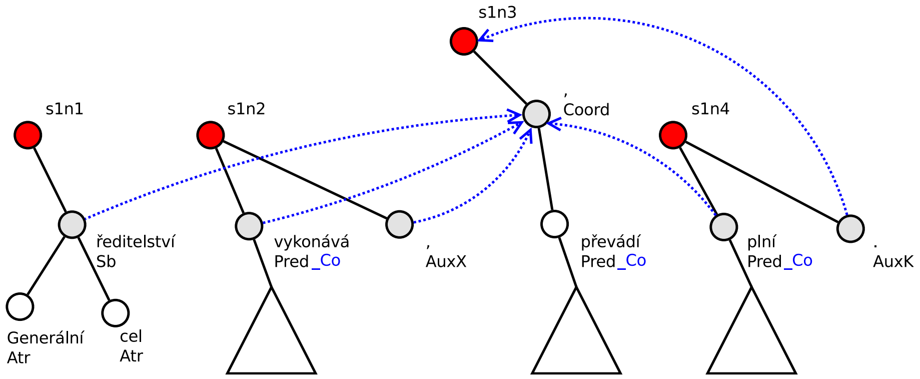


Figure 10: Merging the segment annotations into a dependency tree of the original sentence.

4.4 Evaluation and Discussion

In this section we provide an comparison of the statistical dependency parser and manual annotation. We evaluate both (1) syntactic functions, and (2) dependency relations. We provide the basic statistics in Table 9.

Syntactic function It has been found that there is a substantial difference (over 50% higher, relatively) between the percentage of errors in the segmented sentences vs. function assignment errors in the non-segmented ones. We can see two possible reasons for this results: (i) Segmented *sentences* are not typical sentences. In most cases they represent one member of a coordination. Therefore one or more important afuns may missing, e.g. subject, predicate or object. This may cause problems for parser trained on *full* sentences. (ii) During the manual correction process, annotator assigned afuns as they should be in original, non-segmented sentence. On the other hand, automatic parser assigns afuns locally, without the knowledge of the whole sentence. Table 10 presents the functions in which the parser erred most frequently.

Dependency relations We can see that the overall dependency error rate is visibly higher than has been reported for the newspaper and magazine documents, as reported in (McDonald et al., 2005). However, the parser accuracy (dependency-wise) does not differ as much between the segmented and non-segmented sentences, which is the positive consequence of complex sentence segmentation, leading to similar and reasonable average segment sizes in around 21-25 words per segment.

Parser errors	Segments	Non-segmented sentences	Error rate
Dep's	24.5 %	18.4 %	20.0 %
Func's	23.8 %	12.1 %	15.2 %
Nodes	9,198	25,887	35,085
Segments	536	1,032	1,568
Sent's	101	1,032	1,133

Table 9: Difference in error rates in the segmented vs. non-segmented sentences

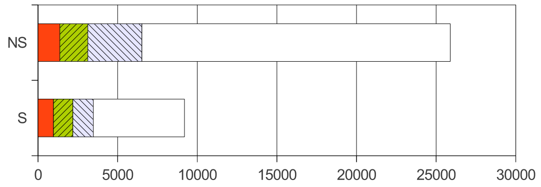


Figure 11: Red - both afun and dependency incorrectly. Green - afun incorrectly and dependency correctly. Blue - afun correctly and dependency incorrectly. White - both afun and dependency correctly

afun	# of errors	% of errors
Atr	1149	21.6%
Obj	896	16.8%
Adv	889	16.7%
graphical	552	10.4%
Sb	457	8.6%
other	1376	25.9%
Total	5319	100.0%

Table 10: Parser errors in assigning dependency functions

5 Future Work

In this section we list our future plans for our sub-tasks: (i) the sub-task of detecting semantic relations in legal documents; and (ii) the issue of parsing of Czech legal texts. We consider the sub-task of detecting references in court decisions to be closed.

5.1 Extracting Knowledge from Unstructured Texts

We used Czech legal texts to demonstrate the features of the RExtractor system. To present the RExtractor language independence we implemented the extraction strategy for English legal texts as well. In general, adding a new language means (i) to use NLP tools for the target language; and (ii) to define database of entities (DBE) and database of queries (DBR).

We want to demonstrate the system on other domains as well. However, we would like to find real use cases where extracted data will be used in real applications. We have already found such opportunities in the medical domain (extraction information from drugs prescriptions), in the banking domain and in police record processing. In addition, we want to compare our system with other approaches and systems. We provided a research of available benchmarks.

To define the tree queries for relation extraction, an assistance of a PML-TQ expert is needed. To eliminate this obstacle, we want to implement a simple graphical interface where a tree query will be defined automatically based on user textual annotations.

In addition, we will place the emphasis on the evaluation taking into consideration various aspects, mainly gold standard data vs. practical use cases, developers' experience vs. users' expectations, scientific contribution vs. 'making life easier'.

5.2 Dependency Parsing of Czech Legal Texts

Thanks to the manually annotated goldstandard data we are able to evaluate all available parsers and select the one with the best performance, on legal domain. In fact, one could consider the currently presented evaluation as irrelevant, because we used the same parser for initial parsing as for its evaluation. In our annotation strategy, the annotator could miss some errors or accept structures that could be different from structures created from scratch. We plan to annotate a part of the data again by another annotator to get the inter-annotator agreement. We see the issue of dependency parsing as the most interesting and challenging for our future work.

Acknowledgments

I would like to thank Barbora Hladká, my PhD supervisor, for her support.

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 highly appreciate RExtractor-related discussions with Martin Nečaský and colleagues from Sysnet, Ltd.

I gratefully acknowledge support from the Technology Agency of the Czech Republic (grant no. TA02010182) and from SVV project no. 260 224. This work has been using language resources developed and/or stored and/or distributed by the LINDAT/CLARIN project.

References

D. Amerland. 2013. *Google Semantic Search: Search Engine Optimization (SEO) Techniques that Get Your Company More Traffic, Increase Brand Impact and Amplify Your Online Presence*. Que Biz-Tech. Que Pub.

L Bacci, E Francesconi, and MT Sagri. 2012. A rule-based parsing approach for detecting case law references in italian court decisions. In *Semantic Processing of Legal Texts (SPLeT-2012) Workshop Programme*, page 27.

Eduard Bejček, Eva Hajičová, Jan Hajič, Pavlína Jínová, Václava Kettnerová, Veronika Kolářová, Marie Mikulová, Jiří Mírovský, Anna Nedoluzhko, Jarmila Panevová, Lucie Poláková, Magda Ševčíková, Jan Štěpánek, and Šárka Zikánová. 2013. Prague dependency treebank 3.0. http://ufal.mff.cuni.cz/pdt3.0.

Tim Berners-Lee, James Hendler, Ora Lassila, et al. 2001. The semantic web. *Scientific american*, 284(5):28–37.

Chris Biemann. 2005. Ontology learning from text: A survey of methods. In *LDV forum*, volume 20, pages 75–93.

Mírian Bruckschen, Caio Northfleet, DM Silva, Paulo Bridi, Roger Granada, Renata Vieira, Prasad Rao, and Tomas Sander. 2010. Named entity recognition in the legal domain for ontology population. In *Workshop Programme*, page 16.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *AAAI*.

Christian Chiarcos, Sebastian Hellmann, and Sebastian Nordhoff. 2012. Introduction and overview. In Christian Chiarcos, Sebastian Nordhoff, and Sebastian Hellmann, editors, *Linked Data in Linguistics*, pages 1–12. Springer Berlin Heidelberg.

Emile DE, Radboud Winkels, and Tom van Engers. 2006. Automated detection of reference structures in law. *Frontiers in Artificial Intelligence and Applications*, page 41.

Felice Dell'Orletta, Simone Marchi, Simonetta Montemagni, Barbara Plank, and Giulia Venturi. 2012. 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.

Christopher Dozier, Ravikumar Kondadadi, Marc Light, Arun Vachher, Sriharsha Veeramachaneni, and Ramdev Wudali. 2010. Named entity recognition and resolution in legal text. In *Semantic Processing of Legal Texts*, pages 27–43. Springer.

Peter Exner and Pierre Nugues. 2012. Entity extraction: From unstructured text to dbpedia rdf triples.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.

E. Francesconi, S. Montemagni, W. Peters, and D. Tiscornia. 2010. *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*. LNCS sublibrary: Artificial intelligence. Springer.

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, and Magda Ševčíková-Razímová. 2006. Prague dependency treebank 2.0.

Irena Holubová, Tomáš Knap, Vincent Kríž, Martin Nečaský, and Barbora Hladká. 2014. INTLIB - an INTelligent LIBrary. In Karel Richta, Václav Snášel, and Jaroslav Pokorný, editors, *Proceedings of the Dateso 2014 Annual International Workshop on DAtabases, TExts, Specifications and Objects*, pages 13–24, Praha, Czechia. Czech Technical University in Prague, Faculty of Information Technology, Czech Technical University in Prague, Faculty of Information Technology.

Vincent Kríž, Barbora Hladká, Martin Nečaský, and Jan Dědek. 2014a. Statistical recognition of references in czech court decisions. In *13th Mexican International Conference on Artificial Intelligence, MICAI 2014, Tuxtla Gutiérrez, Mexico, November 16-22, 2014. Proceedings, Part I*, volume 8856 of *Lecture Notes in Computer Science*, pages 51–61, Switzerland. Instituto Tecnológico de Tuxtla Gutiérrez, Springer International Publishing.

Vincent Kríž, Barbora Hladká, Martin Nečaský, and Tomáš Knap. 2014b. Data extraction using NLP techniques and its transformation to linked data. In *13th Mexican International Conference on Artificial Intelligence, MICAI 2014, Tuxtla Gutiérrez, Mexico, November 16-22, 2014. Proceedings, Part I*, volume 8856 of *Lecture Notes in Computer Science*, pages 113–124, Switzerland. Instituto Tecnológico de Tuxtla Gutiérrez, Springer International Publishing.

Ora Lassila and Ralph R. Swick. 1999. Resource description framework (RDF) model and syntax specification. Technical report. `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/`.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 28–36, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Langauge Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, BC, Canada. Association for Computational Linguistics, Association for Computational Linguistics.

Bernard Merialdo. 1994. Tagging english text with a probabilistic model. *Comput. Linguist.*, 20(2):155–171, June.

Martin Nečaský, Tomáš Knap, Jakub Klímek, Irena Holubová, and Barbora Hladká. 2013. Linked open data for legislative domain - ontology and experimental data. In *Lecture Notes in Business Information Processing*, pages 172–183, Berlin / Heidelberg. Springer.

Petr Pajas and Jan Štěpánek. 2009. System for querying syntactically annotated corpora. In Gary Lee and Sabine Schulte im Walde, editors, *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pages 33–36, Suntec, Singapore. Association for Computational Linguistics.

Karel Pala, Pavel Rychlý, and Pavel Šmerk. 2010. Automatic identification of legal terms in czech law texts. In *Semantic Processing of Legal Texts*, pages 83–94, Berlin. Springer.

Monica Palmirani, Raffaella Brighi, and Matteo Massini. 2003. Automated extraction of normative references in legal texts. In *Proceedings of the 9th international conference on Artificial intelligence and law*, pages 105–106. ACM.

Martin Popel and Zdeněk Žabokrtský. 2010. TectoMT: Modular NLP framework. In Hrafn Loftsson, Eirikur Rögnvaldsson, and Sigrun Helgadottir, editors, *Lecture Notes in Artificial Intelligence, Proceedings of the 7th International Conference on Advances in Natural Language Processing (IceTAL 2010)*, volume 6233 of *Lecture Notes in Computer Science*, pages 293–304, Berlin / Heidelberg. Iceland Centre for Language Technology (ICLT), Springer.

Paulo Quaresma and Teresa Gonçalves. 2010. Using linguistic information and machine learning techniques to identify entities from juridical documents. In *Semantic Processing of Legal Texts*, pages 44–59. Springer.