# Data Extraction Using NLP Techniques and Its Transformation to Linked Data

Vincent Kríž[1], Barbora Hladká[1], Martin Nečaský[2], and Tomáš Knap[2]

[1] Institute of Formal and Applied Linguistics
[2] Department of Software Engineering
Faculty of Mathematics and Physics, Charles University in Prague
Malostranské nám. 25, 118 00 Praha 1, Czech Republic

**Abstract.** We present a system that extracts a knowledge base from raw unstructured texts that is designed as a set of entities and their relations and represented in an ontological framework. The extraction pipeline processes input texts by linguistically-aware tools and extracts entities and relations from their syntactic representation. Consequently, the extracted data is represented according to the Linked Data principles. The system is designed both domain and language independent and provides users with data for more intelligent search than full-text search. We present our first case study on processing Czech legal texts.

## 1 Introduction

According to the statistics provided by International Data Corporation [1], 90% of all available digital data is unstructured and its amount currently grows twice as fast as structured data. 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 are *full-text search* and *metadata search*. In general, both approaches do not work with the semantic interpretation of documents.

We depict the relationship between the fields of Information Extraction (IE) and Semantic Web (SW) in the scheme displayed in Figure 1 where the components of Gathering data and Linguistic analysis belong to IE and while the components of Data representation and Data linking belong to SW. The components are characterized by general features that are typically domain and language independent. However, their design in an extraction pipeline must take into account specifications related to a domain under consideration.

In our work, we focus on the components of Linguistic analysis and Data representation. We deal with the semantics through a *knowledge base* composed of entities and their relations. The knowledge base is built from raw texts by extraction of entities and relations referring to real-world objects. Namely, we exploit dependency trees where both entities and relations are recognized. The outputs are presented according to the Linked Data Principles[1] in the Resource Description Framework (RDF, [2]) that is, in connection

[1] `http://www.w3.org/wiki/LinkedData`

A. Gelbukh et al. (Eds.): MICAI 2014, Part I, LNAI 8856, pp. 113–124, 2014.

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.

[3] 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 specify the semantic information through exploiting syntactic structures in documents. First, we detect entities in the documents, i.e., in syntactic structures of the sentences present in the document and then we link each entity occurrence with its ontological concept. Second, we detect relations in syntactic structures and we enrich the ontological concepts with formal definitions of entities, rights of entities, and obligations declared in documents. Such information can be useful for various users, e.g., an accountant can easily find the definition of a given accounting term and each occurrence of this term in documents; a patient can browse the insurance act to find his rights; an employer can obtain a list of his obligations to his employees.

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[3] or ZákonyProLidi.cz[4]), the systems do not offer any additional information, for example hyperlinks to refered documents.

Our paper is organized as follows: in Section 2, we provide an overview of works related to our topic. The RExtractor system is a complex system that (i) processes input documents by natural language processing (NLP) tools and (ii) queries linguistic structures to extract entities and their relations. Its architecture and components are described in Section 3. In general, the RExtractor system is designed to be domain independent but some modifications must be done when using it for a specific domain. In Section 4 we present the steps that we undertook during the processing of Czech legal texts. We also include the evaluation of this case study. Once the data from the legal texts is extracted, its representation according to a chosen data model follows. Details on this step are described in Section 5. In Section 6 we provide an outline of our future work that covers both improving syntactic parsing and linking ontological concepts.

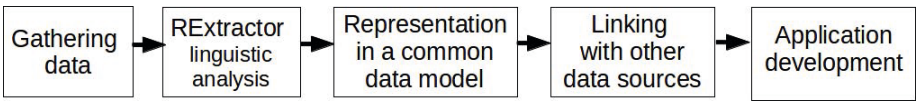


**Fig. 1.** A scheme of data extraction, its representation and exploitation

## 2 Related Work

Works on *relation extraction*: The extraction of relational facts from raw texts has been of interest in information extraction for last decade. With the emergence of the Semantic Web [4] and ontologies [5], data integration has become an additional challenge.

[2] http://www.w3.org/TR/rdf-sparql-query/
[3] http://systemaspi.cz
[4] http://zakonyprolidi.cz

There has been a considerable amount of research on applying semi-supervised methods for data integration [6,7,8]. Unsupervised approaches have contributed further improvements by not requiring hand-labeled data [9,10]. [11] presents SOFIE, a system for automated ontology extension. The system performance was evaluated on the corpus of 150 newspapers articles and the authors report 91.30% precision and 31.08% recall. [12] presents the platform MeTAE. It allows extraction and annotation of medical entities and relationships from medical texts and their representation in the RDF format. They evaluateed the extraction of treatment relations between a treatment (e.g., medication) and an illness (e.g., disease): they obtained 75.72% precision and 60.46% recall.[13] employs a combination of NLP tools, including semantic parsing, coreference resolution, and named entity linking. They proposed an end-to-end system, that extracts entity relations from plain text and attempted to map entities onto the DBpedia namespace. They reported precision of 74.3% and recall of 59.9%. [14] proposes a complex pipeline of NLP tools for Czech performing extraction of basic facts presented in a text. An automatic syntactic analysis is used for extracting phrases that are later classified using Czech WordNet into several semantic categories, such as *Location* or *Time*. They present the results of manual evaluation on 50 randomly selected sentences from internet news groups. They reported accuracy of 69.9%.

At least to our best knowledge, [15] presents the very first results on the legislative domain. The authors implemented two modules to qualify fragments of normative texts in terms of provision types and to extract their arguments. The evaluation set for their extractor consists of 473 law paragraphs. They report accuracy of 82%.

Works on *linked data*: In the proceedings [16], recent relevant developments are documented, mainly language archives for language documentation, typological databases, lexical-semantic resources in NLP, multi-layer annotations and semantic annotation of corpora.

Works on *NLP and the legislative domain*: An elaborated overview of current efforts in legal text processing is given by [17]. The main issues include information extraction, construction of knowledge resources, automatic summarization and translation. [18] showes that the state-of-the-art statistical parser can handle even complex syntactic constructions of an appellate court judge.A few attempts have been carried out to check the performance of parsers on legal texts. One of the main reasons lies in 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 [19] looked at different parsing systems which have been tested against Italian and English legal data sets. All submitted systems concentrated on tuning parameters of machine learning methods they applied.

The processing of Czech legal texts has been overviewed during the work on the Dictionary of law terms [20]. The authors used partial parsing to extract noun groups as the main candidates for legal terms and they explored the valency frames of verbs to link together the established law terms [21]. Processing of non-Czech legal texts is established as well, see e.g. [17] for a review of current efforts.

## 3 RExtractor Architecture

We have proposed a general, domain and language independent architecture called *RExtractor*. The RExtractor system is displayed in Figure 2 and it consists of four components:

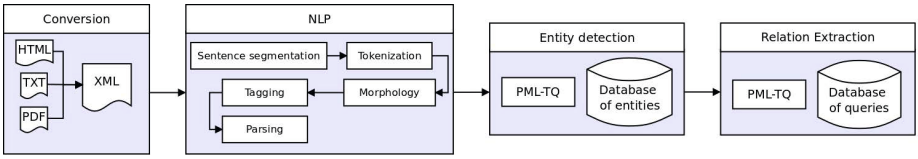


**Fig. 2.** RExtractor architecture

**Conversion** – a largely technical component converting various input formats into internal representation.
**Natural Language Processing** – a linguistic component providing various analyses of input texts, namely sentence segmentation, tokenization, morphological analysis, part-of-speech tagging, and syntactic parsing. Currently employed procedures fit the framework originally formulated in the Prague Dependency Treebank (PDT) [22,23].[5] The dependency approach to syntactic parsing with the main role of verb is applied and it results in a dependency tree where each token in the sentence has one corresponding node and dependencies are assigned with the syntactic dependency relation, as illustrated in Figure 4. The procedures are available in the Treex framework [24].
**Entity Detection** – an extraction component querying dependency trees to detect entities stored in Database of Entities (DBE, see Figure 2) in texts and it exploits the PML-TQ tool [25].[6] DBE is built by domain experts. We prefer querying dependency trees to matching texts with regular expressions because it allows us to detect entities with more complex structures, like coordination. Figure 3 displays the dependency tree of coordination *current tangible and intangible assets*. Using the PML-TQ query displayed in Figure 3, we detect the entity *current tangible assets* in the tree. Because of 1:1 correspondence between tokens in the sentence and nodes in its dependency tree, we can directly mark entities in the original input text.
**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 2). DBS is built by both domain and PML-TQ experts.

For illustration, we assume the sentence *(3) Accounting units, which keep books in simplified extent, create fixed items and reserves according to special legal regulations*, its dependency tree and the query displayed in Figure 4. The query is designed to extract responsibilities of accounting units. Table 1 lists data extracted from the given tree by the given query.

[5] `http://ufal.mff.cuni.cz/pdt3.0/`
[6] `http://ufal.mff.cuni.cz/pmltq/`

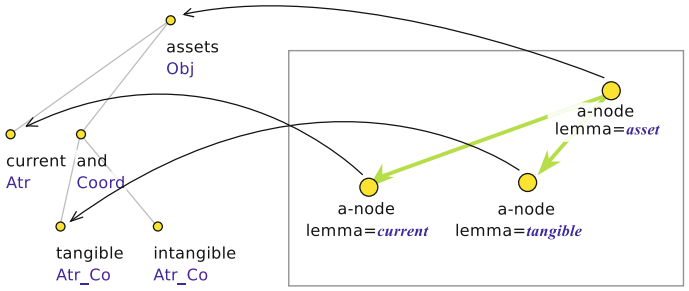


**Fig. 3.** Dependency tree of coordination *current tangible and intangible assets* and tree query for detection of the entity *current tangible assets*

**Table 1.** Data extracted by the query from the tree in Figure 4

| **Subject** | **Predicate** | **Object** |
|---|---|---|
| *Entity* | *hasToCreate* | *Something* |
| `id:1`<br>Accounting units<br>Účetní jednotky | `id:6`<br>create<br>tvoří | `id:3`<br>fixed items<br>opravné položky |
| `id:1`<br>Accounting units<br>Účetní jednotky | `id:6`<br>create<br>tvoří | `id:4`<br>reserves<br>rezervy |

## 4 RExtractor on Czech Legal Texts

In the pilot study, we used RExtractor for processing acts, decrees, and regulations published in the Collection of Laws of the Czech Republic. We list specifics related to the legislative domain for each RExtractor component.

**Conversion.** Although legal texts under consideration have strictly hierarchical structure, there is no official machine readable source of them. Therefore, we converted the input texts according to the RExtractor XML Schema.

**Natural Language Processing.** 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. 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. However, the style of legal texts is "generally considered very difficult to read and understand". [26]

We can see the syntactic parsing as a key procedure employed in RExtractor. However, NLP procedures we have at our disposal for Czech are trained on newspaper texts.[7]

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

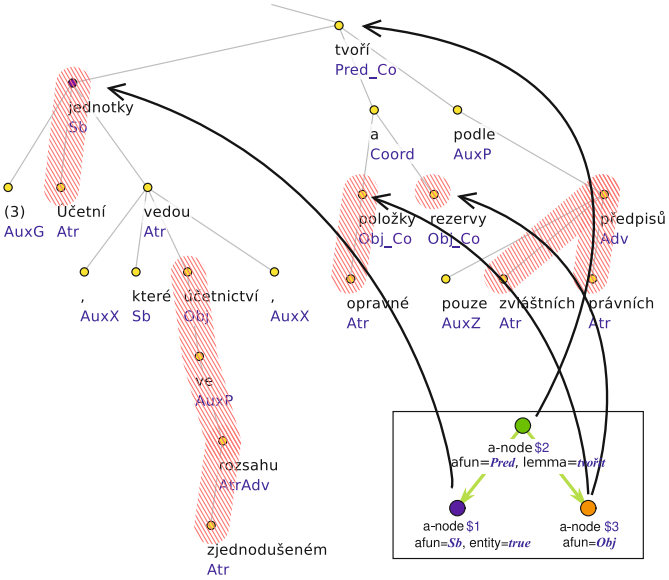


**Fig. 4.** Query matching in a dependency tree

Since legal texts and newspaper texts essentially differ in syntactic features, special attention must be payed to the verification whether we can use the parser trained on newspaper texts or some modifications are needed. To address this issue, we use the manually annotated Corpus of Czech legal texts (CCLT, [27]) consisting of two legal documents from the Collection of Laws of the Czech Republic.[8] The corpus contains 1,133 manually annotated dependency trees with 35,085 nodes in total. The document selection was motivated in the wider context of the INTLIB project.[9]

We have already implemented two preprocessing steps which potentially could improve parser performance, namely sentence splitting and re-tokenization. Both manual and automatic parsing become more difficult with the higher number of words in a sentence. Thus we split long sentences occurring in lists into several shorter sentences, see Table 2. In addition, we adopt the idea of re-tokenization [27] – joining several tokens into one token – that implies reduction of nodes in a tree.

**Entity Detection.** Entities in the decree from CCLT were manually recognized by accounting experts and automatically parsed to import their dependency trees into DBE. Consequently, the queries for entity detection were automatically generated from these trees.

Since CCLT is manually syntactically annotated, we evaluated the process of entity detection against it. We automatically parsed the decree, queryied its manual and automatic trees and compare the extracted entities. Table 3 presents the results. One can observe relatively low precision which is caused by the high number of false positives.

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

[9] `http://ufal.mff.cuni.cz/intlib`

**Table 2.** Long lists and enumerations replaced with several shorter sentences

| Input sentence | Output sentences |
|---|---|
| (1) The General Directorate of Customs | |
| a) is an administrative ... | The General Directorate of Customs is an administrative ... |
| b) administers the customs ... | The General Directorate of Customs administers the customs ... |
| c) functions as a ... | The General Directorate of Customs functions as ... |

It means that RExtractor detected entities which were not annotated manually in goldstandard data. We investigated, that almost in most of such cases, the goldstandard annotation is missing. The human annotators were not consistent and they did not annotate each occurrence of a given entity. Other conclusion from the experiment is positive and it says that the parser has very low influence on the performance of entity detection.

**Relation Extraction** 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 should respect the strategy to cover the maximum number of relations with the minimum number of queries.

In the pilot study, we used the act from CCLT for the query development. Finally, we obtained 5 queries for Definitions, 4 queries for Rights and 2 queries for Obligations. A sample of the queries is presented in Table 7. We carried out the evaluation on the decree from CCLT where we manually detected relations. The decree consists of 762 sentences, 21,967 tokens and 467 relations. We compared them to the queries output and we obtained the results presented in Table 4: the row *Goldstandard* lists the number of manually detected relations. The next rows present the RExtractor output. We determined three types of errors for incorrectly detected relation, see false negatives and false positives in Table 4): (i) incorrect dependency tree, (ii) missing or incorrect query, (iii) missing or incorrect entity. The results are summarized in Table 6.

We collected a set of 28 laws on accounting and taxes provisions consisting of 27,808 sentences and 745,137 tokens. We run RExtractor on this collection and we obtained 2,645 relations in total, details are listed in Table 5.

**Table 3.** Evaluation of Entity Detection Component

| Entity Parsing | Extracted | True positives | False positives | False negatives | Precision | Recall |
|---|---|---|---|---|---|---|
| Manual | 16,428 | 9,549 | 6,879 | 628 | 58.1% | 93.8% |
| Automatic | 16,160 | 9,278 | 6,882 | 838 | 57.4% | 91.7% |

**Table 4.** Evaluation of Relation Extraction Component

| | D | O | R | Total |
|---|---|---|---|---|
| # of queries | 5 | 4 | 2 | 11 |
| Goldstandard | 97 | 308 | 62 | 467 |
| Extracted | 70 | 255 | 41 | 366 |
| True positive | 53 | 206 | 36 | 295 |
| False negative | 44 | 102 | 26 | 172 |
| False positive | 17 | 49 | 5 | 71 |
| Precision (%) | 75.7 | 80.8 | 87.8 | 80.6 |
| Recall (%) | 54.6 | 66.9 | 58.1 | 63.2 |

**Table 5.** Number of relations extracted by tree queries from the collection of 28 laws

| D | | R | | O | |
|---|---|---|---|---|---|
| $D_1$ | 36 | $R_1$ | 240 | $O_1$ | 183 |
| $D_2$ | 287 | $R_2$ | 470 | $O_2$ | 37 |
| $D_3$ | 35 | $R_3$ | 127 | | |
| $D_4$ | 466 | $R_4$ | 6 | | |
| $D_5$ | 46 | | | | |
| Total | 1580 | Total | 843 | Total | 220 |

**Table 6.** Error analysis of incorrectly detected relations

| Error | # of errors | Ratio |
|---|---|---|
| Parser | 145 | 59.7 % |
| Query | 93 | 38.3 % |
| Entity | 5 | 2.1 % |

**Table 7.** Simplyfied versions of the most successfull queries. In a PML-TQ query, both subject and object depend on predicate

| Query | Subject | Predicate | Object |
|---|---|---|---|
| $D_4$ | CASE = 7 | LEMMA = rozumět_se | POS = noun, CASE = 1 |
| $R_2$ | AFUN = Sb | LEMMA = odpovídat | LEMMA = za |
| $O_1$ | ENTITY = true | LEMMA = moci | AFUN = Obj, POS = verb |

## 5 RDF Representation of the Data from RExtractor

In our previous work [3], we presented the ontology for representing acts and consolidated expressions in RDF. The ontology represents each act and its consolidated expressions as an RDF resource which can be linked from other data sources according to the Linked Data principles. The ontology also considers representation of act sections and their consolidated expressions. Therefore, each section of each act is also an RDF resource. We considered only the structure of acts, i.e. their sections and links to those sections. However, we did not consider the semantics of acts, i.e. entities and relations between them defined in acts. Now, since we work with RExtractor, we extend our previously published ontology with new components to represent data extracted by RExtractor in RDF.

The extension has two parts. We describe each as a separate ontology. The first one is called *Legal Concepts Ontology*. Its URI is `http://purl.org/lex/ontology/concepts#` and we use a prefix `lexc:` to refer to it in this paper. The ontology enables to represent the extracted entities and relationships between them independently of the original text of the ontology. The classes and predicates introduced by the ontology are depicted in Figure 5.

The core class of the ontology is the class `Concept` whose instances represent the entities extracted by RExtractor. We call those instances *concepts*. A concept defined by an act exists independently of particular versions (consolidated expressions) of the act. However, because the act exists in one or more versions, there are also respective

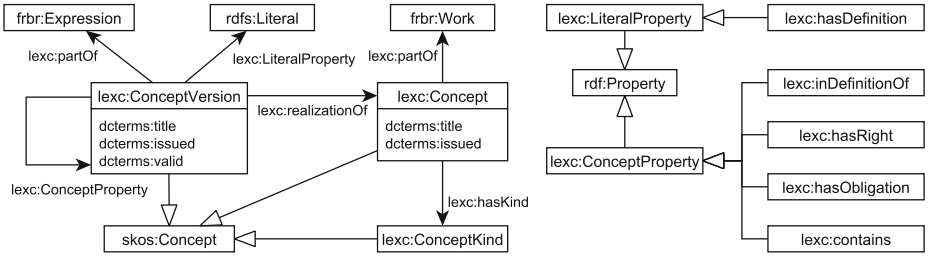


**Fig. 5.** Legal Concepts Ontology

versions of concepts defined by the act. There is a version in which the act defines a concept for the first time. The concept then exists in the following versions of the act until it is cancelled. For each following version, there is a respective version of the concept. Therefore, for each version of the act which speaks about the concept, we also create an instance of the class `ConceptVersion`. This instance represents a particular version of the concept defined by the respective version of the act. Each concept is linked to the act and its sections.This enables us to show users the list of concepts which appear in a chosen act or in any of its sections.

The extracted relations between concepts and their literal properties are represented with sub-properties of the abstract `ConceptProperty` and `LiteralProperty` properties, respectively. However, because each relationship and literal property is extracted from a particular version of an act, the domain of those properties is not the class `Concept` but the class `ConceptVersion`. As Figure 5 shows, there are various sub-properties of the abstract properties and it is easy to add new properties.Currently, there is a literal property `hasDefinition`, which enables to link a concept to its literal definition, and concept properties `hasRight`, `hasObligation`, `inDefinitionOf`, and `contains`, which enable to link a concept to another concept which is the right or obligation of the concept, or is contained in the definition of the concept, or is a part of that concept, respectively.

The `lexc:` ontology enables us to search for literal or concept properties. However, it is not possible to show users the original text of the consolidated act from which a property was extracted by RExtractor. This is very important for users because even precision and recall of RExtractor are relatively high, they are not perfect. Showing the extracted information out of the original textual context could be, therefore, insufficient. Thus, we provide the second extension that we call *Lingvistic Ontology*. Its URI is `http://purl.org/lingv/ontology#` and we use a prefix `lingv:` to refer to it in this paper. The classes and predicates defined by the ontology are depicted in Figure 6.

The core class of the ontology is the class `TextChunk`. It represents a part of the original text (called text chunk) which is the occurrence of some entity (see the sub-class `NamedEntityOccurrence`) or the occurrence of a relationship specification (see the sub-class `RelationOccurrence`). Each text chunk is annotated by its meaning which is a version of some concept (an instance of the class `ConceptVersion` from `lexc:` ontology), relationship between two concepts (a sub-property of `ConceptProperty` from `lexc:` ontology), or literal property (a sub-property of

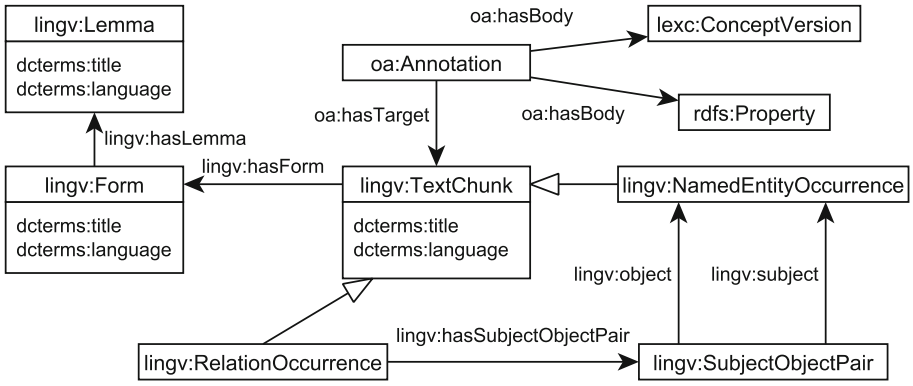


**Fig. 6.** Lingvistic Ontology

`LiteralProperty` from `lexc:` ontology). For representing annotations in RDF we use the Open Annotation Ontology (we use prefix `oa:`).[10]

The `lingv:` ontology enables us to display users text chunks from which RExtractor extracted particular concepts and relations between them. Because a text chunk is also a part of the original text, we are able to show users each text chunk in the context of original documents.

In our experiment we converted the results of RExtractor presented in Table 5 to RDF. The numbers of instances of the main classes from `lexc:` and `lingv:` ontologies are displayed in Table 8.

**Table 8.** The numbers of instances of the main classes from `lexc:` and `lingv:` ontologies

| Class or property | Number of instances |
|---|---|
| `lexc:ConceptVersion` | 3504 |
| `lexc:hasDefinition` | 727 |
| `lexc:hasObligation` | 546 |
| `lexc:hasRight` | 160 |
| `lingv:TextChunk` | 33086 |
| `lingv:NamedEntityOccurrence` | 23674 |
| `lingv:RelationOccurrence` | 1605 |
| `oa:Annotation` | 30800 |

## 6 Conclusion and Future Work

In this paper, we presented a general pipeline of tools for extraction and representation of data that is presented in raw texts. The extraction pipeline processes input texts by linguistically-aware tools and extracts entities and relations from their syntactic representation. Consequently, the extracted data is represented according to the Linked

[10] `http://www.openannotation.org/spec/core/`

Data principles. We applied the tools on texts from the legislative domain. Based on the experience that we acquired in the pilot study, we formulate topics to address in the future:

- improve RExtractor, in particular syntactic parsing and relation query development.
- improve linking of concepts of particular sections of acts to other data sources (e.g., life situations, agendas of public bodies, fines imposed by public bodies, etc.).
- develop web applications which enable users to work with the extracted concepts and relationships and to explore links between extracted concepts and other data sources.

In addition, we will place the emphasis on the evaluation considering a number of aspects, mainly gold standard data vs. practical use cases, developers' experience vs. users' expectations, scientific contribution vs. 'making life easier'.

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

## References

1. Gantz, J., Reinsel, D.: The digital universe decade - are you ready? (2010), http://goo.gl/ZaO0PR
2. Lassila, O., Swick, R.R.: Resource description framework (RDF) model and syntax specification. Technical report (1999), http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/
3. Nečaský, M., Knap, T., Klímek, J., Holubová, I., Vidová-Hladká, B.: Linked open data for legislative domain - ontology and experimental data. In: Abramowicz, W. (ed.) BIS Workshops 2013. LNBIP, vol. 160, pp. 172–183. Springer, Heidelberg (2013)
4. Berners-Lee, T., Hendler, J., Lassila, O., et al.: The semantic web. Scientific American 284, 28–37 (2001)
5. Biemann, C.: Ontology learning from text: A survey of methods. In: LDV forum, vol. 20, pp. 75–93 (2005)
6. Agichtein, E., Gravano, L.: Snowball: Extracting relations from large plain-text collections. In: Proceedings of the Fifth ACM Conference on Digital Libraries, DL 2000, pp. 85–94. ACM, New York (2000)
7. Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Web-scale information extraction in Knowitall (preliminary results). In: Proceedings of the 13th International Conference on World Wide Web, WWW 2004, pp. 100–110. ACM, New York (2004)
8. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr, E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: AAAI (2010)
9. Banko, M., Etzioni, O.: Strategies for lifelong knowledge extraction from the web. In: Proceedings of the 4th International Conference on Knowledge Capture, K-CAP 2007, pp. 95–102. ACM, New York (2007)
10. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1535–1545. Association for Computational Linguistics (2011)

11. Suchanek, F.M., Sozio, M., Weikum, G.: Sofie: a self-organizing framework for information extraction. In: Proceedings of the 18th International Conference on World Wide Web, pp. 631–640. ACM (2009)
12. Abacha, A.B., Zweigenbaum, P.: Automatic extraction of semantic relations between medical entities: a rule based approach. J. Biomedical Semantics 2, S4 (2011)
13. Exner, P., Nugues, P.: Entity extraction: From unstructured text to dbpedia rdf triples. In: The Web of Linked Entities Workshop, WoLE 2012 (2012)
14. Baisa, V., Kovář, V.: Information extraction for czech based on syntactic analysis. In: Vetulani, Z. (ed.) Proceedings of 5th Language and Technology Conference on Human Language Technologies as a Challenge for Computer Science and Linguistics, Pozna, Funcacja Universytetu im. A. Mickiewicza, pp. 466–470 (2011)
15. Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S., Soria, C.: Automatic semantics extraction in law documents. In: Proceedings of the 10th International Conference on Artificial Intelligence and Law, pp. 133–140. ACM (2005)
16. Chiarcos, C., Hellmann, S., Nordhoff, S.: Introduction and overview. In: Chiarcos, C., Nordhoff, S., Hellmann, S. (eds.) Linked Data in Linguistics, pp. 1–12. Springer, Heidelberg (2012)
17. Francesconi, E., Montemagni, S., Peters, W., Tiscornia, D. (eds.): Semantic Processing of Legal Texts. LNCS, vol. 6036. Springer, Heidelberg (2010)
18. McCarty, L.T.: Deep semantic interpretations of legal texts. In: Proceedings of the 11th International Conference on Artificial Intelligence and Law, ICAIL 2007, pp. 217–224. ACM, New York (2007)
19. Dell'Orletta, F., Marchi, S., Montemagni, S., Plank, B., 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)
20. Pala, K., Rychlý, P., Šmerk, P.: Automatic identification of legal terms in czech law texts. In: Semantic Processing of Legal Texts, pp. 83–94. Springer, Berlin (2010)
21. Pala, K., Mráková, E.: Legal terms and word sketches: a case study. In: Sojka, P., Horák, A. (eds.) Proceedings of Fourth Workshop on Recent Advances in Slavonic Natural Languages Processing, RASLAN 2010, Brno, Tribun s.r.o, pp. 31–39 (2010)
22. Hajič, J., Panevová, J., Hajičová, E., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M., Žabokrtský, Z., Ševčíková-Razímová, M.: Prague dependency treebank 2.0 (2006)
23. Bejček, E., Hajičová, E., Hajič, J., Jínová, P., Kettnerová, V., Kolářová, V., Mikulová, M., Mírovský, J., Nedoluzhko, A., Panevová, J., Poláková, L., Ševčíková, M., Štěpánek, J., Zikánová, Š.: Prague dependency treebank 3.0. (2013), http://ufal.mff.cuni.cz/pdt3.0
24. Popel, M., Žabokrtský, Z.: TectoMT: Modular NLP framework. In: Loftsson, H., Rögnvaldsson, E., Helgadóttir, S. (eds.) IceTAL 2010. LNCS, vol. 6233, pp. 293–304. Springer, Heidelberg (2010)
25. Pajas, P., Štěpánek, J.: System for querying syntactically annotated corpora. In: Lee, G., Im Walde, S.S. (eds.) Proceedings of the ACL-IJCNLP 2009 Software Demonstrations, pp. 33–36. Association for Computational Linguistics, Suntec (2009)
26. Tiersma, P.: The Creation, Structure, and Interpretation of the Legal Text (2010), http://www.languageandlaw.org/LEGALTEXT.HTM
27. Kríž, V.: Detecting semantic relations in texts and their integration with external data resources. In: WDS 2013 Proceedings of Contributed Papers, Praha, Czechia, pp. 18–23. Matematicko-fyzikální fakulta Univerzity Karlovy, Matfyzpress (2013)