Intelligent Information Systems 2008
ISBN 978-83-60434-44-4, pages 239–247

Exploitation of linguistic tools in semantic extraction – a design

Jan Dědek[1] and Peter Vojtáš[1,2]

[1] Department of Software Engineering, Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic
[2] Institute of Computer Science, Academy of Sciences of the Czech Republic

Abstract

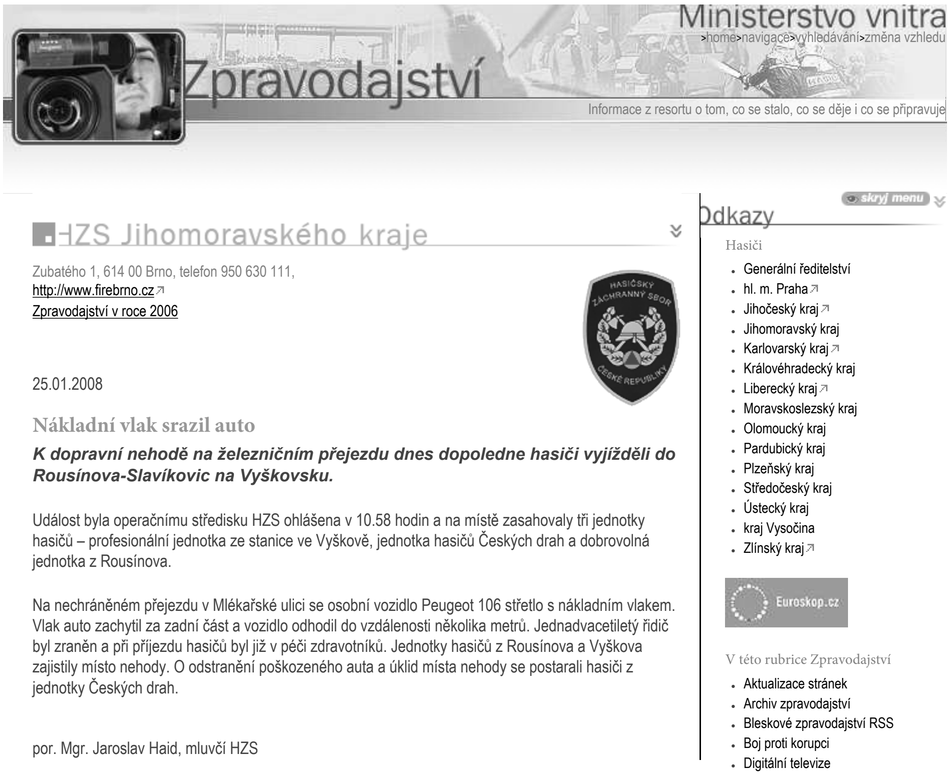
The paper addresses a problem of information extraction from Czech texts from the Web. The method described in the paper exploits existing linguistic tools created originally for a syntactically annotated corpus, Prague Dependency Treebank (PDT 2.0). We propose a system which captures text of web-pages, annotates it linguistically by PDT tools, extracts data and stores the data in an ontology. We report on initial experiments in the domain of reports of traffic accidents. These experiments are promising, e.g. enabling summarization of the number of injured people.

Keywords: information extraction, linguistic annotation, semantic extraction

1 Introduction

The goal of the Semantic Web initiative (Berners-Lee *et al.*, 2001) is to create an universal medium for the exchange of data. It is envisaged to smoothly interconnect systems, integrate application and to support the global sharing of data. The main step is to put machine-understandable data to the Web. To achieve this, we need standards and semantics for semantic enrichment of data. This is well in progress in ontological modeling (see W3C 2004). Further we need to create semantic data, i.e. data annotated by ontological concepts sufficient for machine processing. Semantic annotation of data can be done either during the creation of data (by author or data generator) or after it (third party annotation). The third party annotation assumes we can extract (recognize) data from Web resources.

Semantic extraction of data from Web resources can be seen in three dimensions. The first dimension is the amount of human work associated with the extraction method. This dimension ranges from human hand written through semiautomatic (human trained) to fully automatic approach. The second dimension is the difficulty of the automatic processing of the resource — ranging from easy (tables generated from databases) through semistructured resources to textual resources. The third dimension of the problem is the selectivity of the extraction task — ranging from document classification thorough data region and data record recognition to the extraction of attribute values (Chang *et al.*, 2006). Long distance goal of our research is to come closer to data extraction in the most



HZS Jihomoravského kraje

Zubatého 1, 614 00 Brno, telefon 950 630 111,
http://www.firebrno.cz
Zpravodajství v roce 2006

25.01.2008

Nákladní vlak srazil auto

K dopravní nehodě na železničním přejezdu dnes dopoledne hasiči vyjížděli do Rousínova-Slavíkovic na Vyškovsku.

Událost byla operačnímu středisku HZS ohlášena v 10.58 hodin a na místě zasahovaly tři jednotky hasičů – profesionální jednotka ze stanice ve Vyškově, jednotka hasičů Českých drah a dobrovolná jednotka z Rousínova.

Na nechráněném přejezdu v Mlékařské ulici se osobní vozidlo Peugeot 106 střetlo s nákladním vlakem. Vlak auto zachytil za zadní část a vozidlo odhodil do vzdálenosti několika metrů. Jednadvacetiletý řidič byl zraněn a při příjezdu hasičů byl již v péči zdravotníků. Jednotky hasičů z Rousínova a Vyškova zajistily místo nehody. O odstranění poškozeného auta a úklid místa nehody se postarali hasiči z jednotky Českých drah.

por. Mgr. Jaroslav Haid, mluvčí HZS

Figure 1: Example of the web-page with a report of a fire department

difficult setting in all the dimensions. Some initial proposals and experiments are presented in this paper.

In this paper we describe initial experiments with information extraction from traffic accident reports of fire departments in several regions of the Czech Republic. We would like to demonstrate the prospects of using linguistic tools developed in the Institute of Formal and Applied Linguistics in Prague (described in 3). These experiments are promising, they e.g. enable the summarization of the number of injured people.

1.1 Motivation

The Ministry of Interior of the Czech Republic presents on its Web pages[1] also reports from fire departments of several regions of the Czech Republic. These departments are responsible for rescue and recovery after traffic accidents. These reports are rich in information, e.g. where and when an traffic accident occurred, which units helped, how much time it took them to show up on the place of accident, how many people were injured, killed etc. An example of such report

[1] http://www.mvcr.cz/rss/regionhzs.html

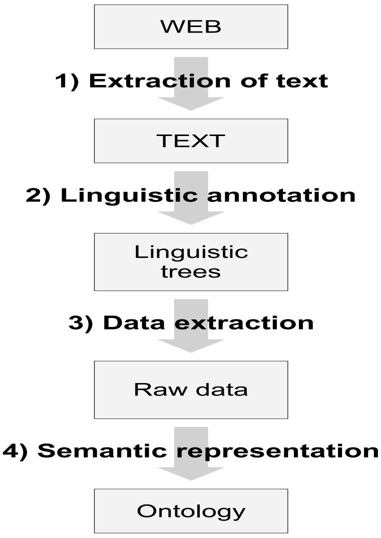


Figure 2: Schema of the extraction process

can be seen in the Figure 1.

Nevertheless, there are also reports about fire accidents and also about fire-fighting contests. The task to extract information on the number of injured/killed people in traffic accidents needs deep linguistic analysis.

In an ideal case all these reports would be already semantically annotated, e.g. using RDFa (W3C, 2007). In this case extraction could be done fully automatically by a software agent. In our case, there is no semantic annotation present in the reports.

2 Semantic extraction

We propose a relatively straightforward process for the extraction of semantic data from text-based web-resources. This process consists of four steps. The Figure 2 describes it.

1. **Extraction of text**
   The linguistic annotating tools process plain text only. In this phase we have to extract the text from the structure of a given web-resource. In this first phase we have used RSS feed of the fire department web-page. From this we have obtained URLs of particular articles and we have downloaded them. Finally we have extracted the desired text by means of a regular expression.
2. **Linguistic annotation**
   In this phase the linguistic annotators process the extracted text and produce corresponding set of dependency trees representing the deep syntactic structure of individual sentences. We have used the linguistic tools described in the section 3 for this task.

3. **Data extraction**
   We use the structure of tectogrammatic (i.e. deep syntactic) dependency trees to extract relevant data. See section 4 for more details.
4. **Semantic representation**
   This phase consists of quite simple data transformation or conversion to the desired ontology format. But it is quite important to choose suitable ontology that will properly represent semantics of the data. We have not implemented this phase yet.

3 Linguistic tools for automatic annotation of texts

In this section we will describe the linguistic tools that we have used to produce linguistic annotation of texts. These tools are being developed in the Institute of Formal and Applied Linguistics[2] in Prague (ÚFAL), Czech Republic. They are publicly available – they have been published on a CD-ROM under the title PDT 2.0 – Hajič *et al.* 2006 (first five tools) and in Klimeš 2006 (Tectogrammatical analysis). These tools are used as a processing chain and at the end of the chain they produce tectogrammatical (Mikulová *et al.*, 2006) dependency trees. The Table 1 shows some details about these tools.

1. **Segmentation and tokenization** consists of tokenization (dividing the input text into words and punctuation) and segmentation (dividing a sequences of tokens into sentences).
2. **Morphological analysis** assigns all possible lemmas and morphological tags to particular word forms (word occurrences) in the text.
3. **Morphological tagging** consists in selecting a single pair lemma-tag from all possible alternatives assigned by the morphological analyzer.
4. **Collins' parser – Czech adaptation** (Collins *et al.*, 1999)
   Unlike the usual approaches to the description of English syntax, the Czech syntactic descriptions are dependency-based, which means, that every edge of a syntactic tree captures the relation of dependency between a governor and its dependent node. Collins' parser gives the most probable parse of a given input sentence.
5. **Analytical function assignment** assigns a description (*analytical function* – in linguistic sense) to every edge in the syntactic (dependency) tree.
6. **Tectogrammatical analysis** (Klimeš, 2006) produces linguistic annotation at the tectogrammatical level.

Although all tools mentioned above are a part of the processing chain mentioned in the previous chapter, the errors they produce are not multiplied. The numbers listed in the table are measured against a corresponding level of the Prague Dependency Treebank. If, for example, the morphological tagger, which is used as a prerequisite of the analytical parser (Collins' parser), has a precision of 93%, the 7% of incorrectly assigned tags definitely have some influence on the quality of the result of the parser. This influence is nevertheless already reflected in the

[2] `http://ufal.mff.cuni.cz`

Table 1: Linguistic tools for machine annotation

Name of the tool	Evaluation results (proclaimed by authors)
Segmentation and tokenization	precision(p): 98,0%, recall(r): 91,4%
Morphological analysis	2,5% unrecognized words
Morphological tagging	93,0% of tags assigned correctly
Collins' parser – Czech adaptation	81,6% dependencies assigned correctly
Analytical function assignment	precision: 92%
Tectogrammatical analysis (Klimeš, 2006)	dependencies p: 90,2%, r: 87,9% assignment of f-tags p: 86,5%, r: 84,3%

81,6% precision of the parser. The results of the parser are measured against the correct trees contained in the PDT, therefore the precision of the parser is the **total** precision of the combination of the tagger and the parser. On the other hand, the (linguistic) analytical function assignment and the tectogrammatical analysis are both more or less dependent on the results of the previous phases and their precision has to be taken as a precision obtained on (already slightly imprecise) results of the analytical parser.

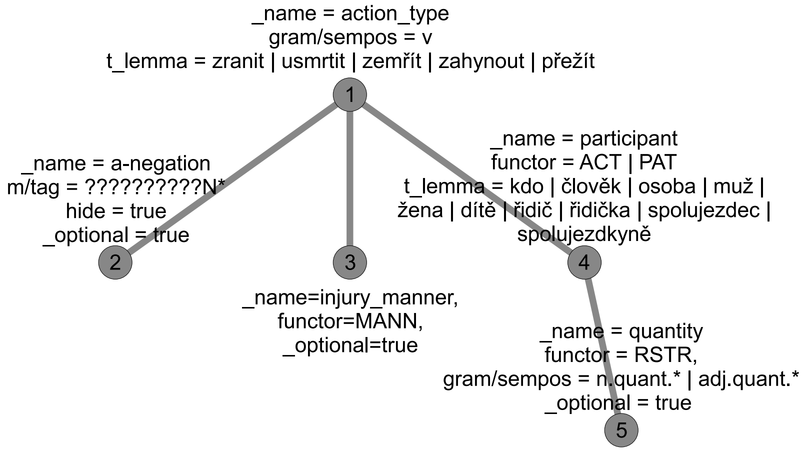
These facts are important especially wrt. possible extension of the system in the future. It might be an interesting research topic to compare the results of the information extraction obtained through the exploitation of the full processing chain versus the results of the analytical parser only - the complications caused by the differences between the analytical level of representation and the desired structure of extracted information might be compensated by decreasing the amount of imprecision present in the system.

4 The data extraction and syntactic structures

The extraction method we have used is based on extraction rules. These rules correspond to query requests of Netgraph application. The Netgraph application (Mírovský, 2006) is a linguistic tool used for searching through a syntactically annotated corpus of a natural language. It was originally developed for searching the analytical and tectogrammatical levels of the Prague Dependency Treebank, a richly syntactically annotated corpus of Czech (Hajič *et al.*, 2006).

Netgraph queries are written in a special query language. An example of such Netgraph query can be found in the Figure 3. The Netgraph is a general tool for searching trees, it is not limited only to the trees in the PDT format. In our application we use it for searching the tectogrammatical trees provided by a set of language processing tools described in the previous chapter. The tectogrammatical trees have a very convenient property of containing just the type of information we need for our purpose, namely the information about inner participants of verbs - actor, patient, addressee etc.

The tectogrammatical (deep syntactic) level of representation is more suitable for our purpose than the analytical (surface syntactic) level of representation of the



Transcript:

zranit to injure	usmrtit to kill	zemřít to die	zahynout to wane	přežít to survive

kdo somebody	člověk (hu)man	osoba person	muž man	žena woman	dítě child

řidič driver	řidička woman driver	spolujezdec passenger	spolujezdkyně woman passenger

Figure 3: Netgraph query – extract rule.

structure of each sentence. The inner participants (actor, patient, addressee etc.) provide much more reliable information about the actual meaning of the sentence than the syntactic roles (subject, object etc.). The roles of a subject or object are misleading especially in the case of passive sentences, where usually the subject of the sentence corresponds to the patient or addressee while the actor is expressed by an object of the passive sentence. In these cases it would be necessary to develop (for the analytical level representation) some kind of algorithm analyzing these cases and providing the assignment of proper roles to individual words, while at the tectogrammatical level we get the desired information directly.

The extraction works as follows: the extraction rule is in the first step evaluated by searching through a set of syntactic trees. Matching trees are returned and the desired information is taken from particular tree nodes.

Let us explain it in more detail by using the example of extraction rule from the Figure 3. This rule consists of five nodes. Each node of the rule will match some node in each matching tree. So we can investigate the relevant information by reading values of linguistic tags of matching nodes. We can find out the number (node number 5) and kind (4) of people, which were or were not (2) killed or injured (1) by an accident that is presented in the given sentence. And we can also identify the manner of injury in the node number 3.

We have evaluated the extraction rule shown in the Figure 3 by using a set of 800 texts of news of several Czech fire departments. There were about 470

```
<injured_result>
      <action type="zranit">
            <sentece>
                  Při požáru byla jedna osoba lehce zraněna -- jednalo se
                  o majitele domu, který si vykloubil rameno.
            </sentece>
            <sentece_id>T-vysocina63466.txt-001-p1s4</sentece_id>
            <negation>false</negation>
            <manner>lehký</manner>
            <participant type="osoba">
                  <quantity>1</quantity>
                  <full_string>jedna osoba</full_string>
            </participant>
      </action>
      <action type="zemřít">
            <sentece>
                  Ve zdemolovaném trabantu na místě zemřeli dva muži -- 82letý
                  senior a další muž, jehož totožnost zjišťují policisté.
            </sentece>
            <sentece_id>T-jihomoravsky49640.txt-001-p1s4</sentece_id>
            <negation>false</negation>
            <participant type="muž">
                  <quantity>2</quantity>
                  <full_string>dva muži</full_string>
            </participant>
      </action>
      <action type="zranit">
            <sentece>Čtyřiatřicetiletý řidič nebyl zraněn.</sentece>
            <sentece_id>T-jihomoravsky49736.txt-001-p4s3</sentece_id>
            <negation>true</negation>
            <participant type="řidič">
                  <full_string>Čtyřiatřicetiletý řidič</full_string>
            </participant>
      </action>
</injured_result>
```

FIGURE 4: Example of the result of the extraction procedure.

sentences matching the rule and we found about 200 numeric values contained in the node number 5. This extraction rule (Figure 3) is a result of a learning procedure of a human designer. We are going to support and automatize the procedure of learning extraction rules in our future work.

4.1 Extraction output

Small part of the result of the extraction is shown in the Figure 4. This result contains three pieces of information extracted from three articles.

Each piece of information is closed in the `<action>` element and each deals with some kind of incident that is described in given article.

The attribute `type` specifies the type of the action. So in the first and in the third case there was somebody injured (*zranit* means to injure in Czech) and in the second case somebody died (*zemřít* means to die in Czech).

The element `<negation>` holds the information about negation of the clause. So we can see that the participant of the third action was **not** injured.

The element `<participant>` contains information about the participants of the action. The attribute `type` specifies the type of the participants and the element

`<quantity>` holds the number of the participants. So in the first action only a single person (*osoba*) was injured. In the second action two men (*muž*) died and in the third action a driver (*řidič*) was not injured.

4.2 Gathering similar words

The Figure 3 shows, that it would be useful to gather words with similar meanings in our extraction rules. For example, the rule in the Figure 3 contains long disjunctions of similar words (nodes with numbers 1 and 4). These disjunctions could be replaced with some kind of expression telling that we are looking for any word from some semantic category (e.g. human beings). For this purpose we wanted to use the Czech WordNet (Pala and Smrž, 2004).

After we have explored the records of the Czech WordNet (CzWN) related to the domain of our interest (car accidents, etc.) we have decided not to involve CzWN in the extraction process. The reason is that the coverage of the vocabulary of our domain is rather poor and the semantic connections of words are sometimes unfortunately missing. But we can supply the missing information to CzWN or we can build up a new domain-specific word-net based on the ground of CzWN.

5 Conclusion

We have presented a proposal of a system for semantic extraction of information from Czech text on Web pages. Our system relies on linguistic annotating tools from ÚFAL and the tree querying tool Netgraph. Our contributions are in fact initial experiments in text extraction from downloaded pages, formulation of a rule-based extraction method and demonstration of semantics of the extracted data.

More details can be found in Dědek 2007. In the future we would like to extend this method by domain oriented lexical net and semiautomatic search for interesting extraction rules.

Acknowledgment: This work was partially supported by Czech projects 1ET100300517 and 1ET100300419.

References

Tim Berners-Lee, James Hendler, and Ora Lassila (2001), The Semantic Web, A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities, *Scientific American*, 284(5):34–43.

Chia-Hui Chang, Mohammed Kayed, Moheb Ramzy Girgis, and Khaled F. Shaalan (2006), A Survey of Web Information Extraction Systems, *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411–1428, ISSN 1041-4347, doi:http://doi.ieeecomputersociety.org/10.1109/TKDE.2006.152.

Michael Collins, Jan Hajič, Eric Brill, Lance Ramshaw, and Christoph Tillmann (1999), A Statistical Parser of Czech, in *Proceedings of 37th ACL Conference*, pp. 505–512, University of Maryland, College Park, USA.

Jan Dědek (2007), *Semantic annotation of data from web resources (in Czech)*, Master's thesis, Faculty of Mathematics and Physics, Charles University, Prague, Czech rep.

Jan Hajič, Eva Hajičová, Jaroslava Hlaváčová, Václav Klimeš, Jiří Mírovský, Petr Pajas, Jan Štěpánek, Barbora Vidová-Hladká, and Zdeněk Žabokrtský (2006), Prague Dependency Treebank 2.0 CD-ROM, Linguistic Data Consortium LDC2006T01, Philadelphia 2006.

Václav Klimeš (2006), Transformation-Based Tectogrammatical Analysis of Czech, in *Proceedings of the 9th International Conference, TSD 2006*, number 4188 in Lecture Notes In Computer Science, pp. 135–142, Springer-Verlag Berlin Heidelberg, ISSN 0302-9743.

Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, and Zdeněk Žabokrtský (2006), Annotation on the tectogrammatical level in the Prague Dependency Treebank. Annotation manual, Technical Report 30, ÚFAL MFF UK, Prague, Czech Rep.

Jiří Mírovský (2006), Netgraph: A Tool for Searching in Prague Dependency Treebank 2.0, in Jan Hajič and Joakim Nivre, editors, *Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories (TLT)*, 5, pp. 211–222, Prague, Czech rep., ISBN 80-239-8009-2.

Karel Pala and Pavel Smrž (2004), Building Czech Wordnet, *Romanian Journal of Information Science and Technology*, 2004(7):79–88, URL `http://www.fit.vutbr.cz/research/view_pub.php?id=7682`.

W3C (2004), OWL Web Ontology Language Guide, URL `http://www.w3.org/TR/owl-guide/`.

W3C (2007), RDFa Primer, URL `http://www.w3.org/TR/xhtml-rdfa-primer/`.