RASLAN 2010
Recent Advances in Slavonic Natural Language Processing

P. Sojka, A. Horák (Eds.)

RASLAN 2010

Recent Advances in Slavonic Natural Language Processing

Fourth Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2010
Karlova Studánka, Czech Republic, December 3–5, 2010
Proceedings

Tribun EU
2011

Proceedings Editors

Petr Sojka
Faculty of Informatics, Masaryk University
Department of Computer Graphics and Design
Botanická 68a
CZ-602 00 Brno, Czech Republic
Email: sojka@fi.muni.cz

Aleš Horák
Faculty of Informatics, Masaryk University
Department of Information Technologies
Botanická 68a
CZ-602 00 Brno, Czech Republic
Email: hales@fi.muni.cz

Katalogizace v knize – Národní knihovna ČR

RASLAN 2010 (4. : Karlova Studánka, Česko)

RASLAN 2010 : Recent Advances in Slavonic Natural Language Processing : fourth workshop on ..., Karlova Studánka, Czech Republic, December 3-5, 2010 : proceeedings / P. Sojka, A. Horák (eds.). - 1st ed. at Tribun EU. - Brno : Tribun EU, 2011. - VIII+115 s.

ISBN 978-80-7399-246-0

81'322 * 004.82/.83:81'322.2
- počítačová lingvistika
- zpracování přirozeného jazyka
- sborníky konferencí

- computational linguistics
- natural language processing
- proceedings of conferences

81 - Lingvistika. Jazyky [11]
004.8 - Umělá inteligence [23]

410 - Linguistics [11]
006.3 - Artificial intelligence [23]

ISBN 978-80-7399-246-0

Preface

This volume contains the Proceedings of the RASLAN (RASLAN 2010) held on December 3rd–5th 2010 in Karlova Studánka, Sporthotel Kurzovní, Jeseníky, Czech Republic.

The RASLAN Workshop is an event dedicated to exchange of information between research teams working on the projects of computer processing of Slavonic languages and related areas going on in the NLP Centre at Faculty of Informatics, Masaryk University, Brno. RASLAN is focused on theoretical as well as technical aspects of the project work, on presentations of verified methods together with descriptions of development trends. The workshop also serves as a place for discussion about new ideas. The intention is to have it as a forum for presentation and discussion of the latest developments in the the field of language engineering, especially for undergraduates and postgraduates affiliated to the NLP Centre at FI MU. We also have to mention the cooperation with the Dept. of Computer Science FEI, VŠB Technical University Ostrava.

Topics of the Workshop include (but are not limited to):

* text corpora and tagging
* syntactic parsing
* sense disambiguation
* machine translation, computer lexicography
* semantic networks and ontologies
* semantic web
* knowledge representation
* logical analysis of natural language
* applied systems and software for NLP

RASLAN 2010 offers a rich program of presentations, short talks, technical papers and mainly discussions. A total of 15 papers were accepted, contributed altogether by 22 authors. Our thanks go to the Program Committee members and we would also like to express our appreciation to all the members of the Organizing Committee for their tireless efforts in organizing the Workshop and ensuring its smooth running. In particular, we would like to mention the work of Pavel Rychlý, Aleš Horák and Dana Hlaváčková. The TEXpertise of Petr Sojka resulted in efficient production of the volume which you are now holding in your hands. Last but not least, the cooperation of Tribun, tribun.eu as both publisher and printer of these proceedings is gratefully acknowledged.

Brno, December 2010 Karel Pala

Table of Contents

IV Language Applications

Part I

Morphology and Lexicon

A New Data Format for Czech Morphological Analysis

Pavel Šmerk

Faculty of Informatics, Masaryk University
Botanická 68a, CZ-60200 Brno, Czech Republic
smerk@mail.muni.cz

Abstract. The paper presents a new data format for computational morphology of Czech. The new format allows for a significant reduction of a redundancy yielded by existing formats. It is also much more linguistically interpretable and acceptable. The paper shows that there is no need to develop any computer-specific description of morphology, but that the traditional linguistic description suffices quite well.

1 Introduction

At the first sight, the morphological analysis and synthesis of Czech seems to be a well-solved task. For more than a decade there are available even two well established and broadly used systems for computational morphology of Czech. One of them is developed in Prague [1,2], the other one in Brno [3,4]. These two systems are completely independent, which means that there are two distinct language data sets which describe Czech morphology, two distinct sets of morphological tags, two data formats, and two analyzers.

Despite of many particular differences, the general principle of the language data description is the same. In both solutions the data consist of so-called paradigms, i. e. sets of word endings and corresponding morphological tags, and of a list of lemmata or word stems. Each word stem is assigned to some paradigm in such a way that concatenations of the stem with the paradigm's endings yield all forms of the word along with appropriate morphological tags. The thing is, the stems and the endings are never modified, but only concatenated during a synthesis or separated from a word form during an analysis.

Such an approach is rather inadequate for a language like Czech which has a rich set of graphemic, phonological and morphological alternations. The problem is that these alternations require to set up distinct paradigms even for words which are inflected quite equally but which differ in some—although completely regular—alternations. For example, surnames *Staněk*, *Hromek*, and *Polák* with genitive singular forms *Staňka*, *Hromka*, and *Poláka* obey exactly the same rules within the inflection, but they have to be described by means of three paradigms which contain endings *něk* and *ňka*, *ek* and *ka*, or *0* and *a* respectively.

As a consequence, the number of the paradigms is very high and the paradigm system is therefore very redundant. This redundancy inevitably leads

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 3–8, 2010.

either to an increase in inconsistencies or even errors in the data, or to a strong need of powerful tools which inhibit emergence of the inconsistency. For a more detailed discussion see [5].

In the following section we offer a proposal of a new data format which lowers the redundancy of the data. Then, in the Section 3, we show results of utilization of the new data format in a description of masculine animate nouns. Finally we sum up the conclusions and sketch out some necessary future work.

2 The New Data Format

As the current data formats do, the new format also divides the data into two parts: a lexicon and paradigms. What is rather new, is the intention to let the lexicon cover the idiosyncracies, whereas the paradigms, and also some rules in the program which interprets the format, should describe only the regularities in the data.

The very basic principle of the data organization remains unchanged: the lexicon contains the stems with names of paradigms, e. g. `slon:pán`, and the paradigms are set of endings and appropriate tags, e. g.

```
pán
        k1gMnSc1        0
        k1gMnSc2        a
        ...
```

The endings are appended to the stems, but as a result, and this is the essential difference, we obtain only structures like `pán-0`, `pán-a`, ... along with tags `k1gMnSc1`, `k1gMnSc2`, ... To derive the "surface" word forms from these structures, some additional rules have to be applied.

Obviously, the most trivial rules have to remove the `-` (which can be interpreted as a morpheme boundary) and `0` (zero ending). Other rules deal with graphemic alternations like `ňe` → `ně`, e. g. `tuleň-e` → `tuleňe` → `tuleně`. Another rules describe the phonological alternations like `k-i` → `c-i`, e. g. `vlk-i` → `vlc-i` → `vlci`. And yet another rules are used to handle some morphological (but in fact phonological as well) alternations like vowels alternating with zero `.VC-0` → `VC-0` and `.VC-V` → `C-V`, e. g. `ďáb.el-a` → `ďábl-a` → `ďábla`.

The paradigm may allow for more than one ending for a particular tag. In such a situation, regular expressions describe a context (possible stem ends) in which the given ending may be used. Omitting the regular expression denotes the default option (an unmarked ending):

```
        k1gMnPc6        ech, ích/[kgc]|ch
```

Even these few above mentioned simple improvements allows us to replace a big portion of the former paradigm system with fewer more general paradigms, but the redundancy would still remain high. To lower it, the new format offers the following possibilities (among others and only in brief):

– the paradigm can be defined as a modification of another paradigm:

```
soudce:muž
        k1gMnPc1        e
        k1gMnPc5        e
```

– inflection of a stem can be described not only by one paradigm, but also by a list of paradigms in which the latter overwrites—or, if the paradigm's name is prepended by a plus sign, is added to—the former;
– the previous has a sense only if the paradigm is allowed to be "incomplete". One can either define a "paradigm" even for a single ending like

```
-ové
        k1gMnPc1        ové
        k1gMnPc5        ové
```

or use a regular expression to select only a subset of endings of a given paradigm, e. g. `pán_nP` selects only endings whose tags contain `nP`. As examples of these posibilities, consider the following lexicon entries:

```
dřevokaz:pán,+muž
Marcel:pán,-ové,muž_nSc5
```

– if a word or its stem has some irregular forms, these forms have to be explicitely listed in the lexicon, e. g.

```
přítel:muž
        přátel:muž_nP,-é
        přátel-0        k1gMnPc2
```

where, again, the more specific overwrites—or is added to—the more general;
– we also need a possibility to describe differences between the written form and pronunciation, especially for words of foreign origin, because the analyser deals with the former, but the inflection is driven by the latter. The format uses the following notation:

```
Smith[t:pán,-ové
        +Smith[s:muž,-ové
```

where the regular expressions, and the rules which derive the word form from the structure "see" the stem-final `t` or `s`, but while deleting the `-`, the whole "pronunciation" part between `[` and `-` is also deleted.

2.1 From the Lexicon with Paradigms to the Lexicon with Features

Up to now, the new format allows us to reproduce the information contained in traditional grammar books quite closely. We can describe the inflection of words by means of the traditional paradigms, eventually with some exceptions, just like the grammar books do. But may be this is not the way people have

organized the language data in their heads. It is unlikely that the speakers deal with any paradigms in such a way that they would have some inventory of stems each one "explicitely" linked to some paradigm. More likely they infer the proper inflectional paradigm from some features or properties of the stem. For instance, the native speakers of Czech know that masculine animate nouns ended up with a hard consonant belong to a "hard" declination. Thus there is no need to have an extra information on the paradigm in the lexicon: such an information would be redundant for these nouns.

To implement this idea we allow for an addition of "implicit" rules like these:

```
[sxz]/qJO       muž,pán_nPc[67],+pán_nPc4
$T\Ka           žena_nS,-ovi,pán_nP,-ové
```

where `$T` is a shorthand for a regular expression which defines hard consonants and `qJO` is a tagset extension which denotes proper names of persons.

On the left side of the rule, there are the conditions which have to be satisfied if the rules are to be applied. The condition can describe either the stem end, or the tag, or both (then the two conditions are separated by a slash '/'). On the right side, there is the list of paradigms which is prepended to the list of paradigms from the lexicon—if they are present, they specify some unusual, non-typical behaviour of the stem.

Then, for example, the following entries in the lexicon

```
Klaus k1gMqJOP
houslista:-i,+-é k1gM
```

can stand for a markedly longer definitions

```
Klaus:muž,pán_nPc[67],+pán_nPc4 k1gMqJOP
houslista:žena_nS,-ovi,pán_nP,-ové,-i,+-é k1gM
```

3 Case Study: Masculine Animate Nouns

As a case study we use the new format for a description of masculine animate nouns. In the old data format, these nouns are described by 217 different paradigms.[1] The Table 1 lists all lexical descriptions which are shared by at least 10 lemmata (the representative is chosen arbitrarily).

Taken as a whole, the figures in the table show that more than 92.3 % of masculine animate nouns can be described only by means of part-of-speech specification, some pieces of semantic information and/or internal (morphotactic) structure.[2]

The description of the new paradigms is 13 times smaller than the equivalent 217 old paradigms — and it is even 24 times smaller, if one does not count definitions shared among different genders or parts-of-speech.

[1] For the sake of completeness it should be stated that one of these old paradigms has not assigned any lemma and the most of the paradigms for surnames are duplicates, i. e. there exists an identical paradigm for non-surnames. [2] E. g. `=an` in `Severo+evrop=an` is a suffix which fully determines the inflection of the word, see [5] for more details.

Table 1. The most frequent descriptions in the dictionary

13,871	69.17	`gaučo k1gM`
2,207	11.01	`Ionesc[ko k1gMqJOP`
1,654	8.25	`Severo+evrop=an`
683	3.41	`Mario k1gMqJO`
440	2.19	`kok.eš:-ové k1gM`
321	1.60	`sob.ěk:-i k1gM`
146	0.73	`uniat:-é k1gM`
90	0.45	`invalida:-é,+-i k1gM`
90	0.45	`košer:+-ové k1gM`
58	0.29	`dutoroh=ý k1gMnP`
52	0.26	`tatí:neskl k1gM`
41	0.20	`pterosaur-us:+-i k1gM`
35	0.17	`v%ol k1gMqA`
22	0.11	`příchoz:muž,-ové`
17	0.08	`Ferrari:neskl k1gMqJOP`
16	0.08	`pán k1gM`
		`pane nSc5`
12	0.06	`Řek k1gMqJN`
10	0.05	`Ciceron k1gMqJO`
		`Cicero nSc1`

4 Conclusions and Future Work

The primary goal of the new format was to significantly reduce the redundancy of the current descriptions of the morphological data, but it has several more advantages:

- the words can be filed under the paradigms found in the traditional grammar books;
- it is easy to handle the graphemic and phonologocal alternations;
- the format allows for much more linguistically acceptable and interpretable description of the data and if the same phenomenon can be described in more than one way, the format even allows us to interpret these descriptions differently;
- it is possible to describe markedness and it is possible to distinguish what is regular or at least typical and what is idiosyncratic or peripheral—and more than that: the idiosyncracy can be described only by means of departures from the rules.

The new format even allows for a description of word formation relations and therefore the morpheme structure of the words, but it is not discussed in this paper (see [5] for more details).

Within the future work, the rest of data is and will be converted to the new format. Then the tougher part of the task will follow: a description of a word formation relations.

Acknowledgements

This work has been partly supported by the Ministry of Education, Youth and Sports of Czech Republic under the project LC536 and within the National Research Programme II project 2C06009, and by the Czech Grant Agency under the project GA 407/07/0679.

References

1. Hajič, J.: Disambiguation of Rich Inflection (Computational Morphology of Czech). Karolinum, Praha (2004).
2. Hlaváčová, J.: Formalizace systému české morfologie s ohledem na automatické zpracování českých textů (Formalization of the Czech Morphology System with Respect to the Automatic Processing of Czech Texts). Ph.D. thesis, Faculty of Arts, Charles University, Praha, Czech Republic (2009).
3. Osolsobě, K.: Algoritmický popis české formální morfologie a strojový slovník češtiny (Algorithmical Description of the Czech Formal Morphology and Machine Dictionary of Czech). Ph.D. thesis, Faculty of Arts, Masaryk University, Brno, Czech Republic (1996).
4. Sedláček, R.: Morphemic Analyser for Czech. Ph.D. thesis, Faculty of Informatics, Masaryk University, Brno, Czech Republic (2005).
5. Šmerk, P.: K počítačové morfologické analýze češtiny (On Computational Morphological Analysis of Czech). Ph.D. thesis, Faculty of Informatics, Masaryk University, Brno, Czech Republic (2010).

Valency Frames of Polysemous and Homonymous Verbs in VerbaLex

Vít Baisa

Natural Language Processing Centre, Faculty of Informatics
Masaryk University, Brno, Czech Republic
xbaisa@fi.muni.cz

Abstract. Verb valency lexicon VerbaLex is one of few language resources which take a local context into account. For each Czech verb with its sense, VerbaLex contains appropriate valency frames (patterns with morphological and syntactical information) in which the verb can appear. This, and the fact that a context of a word is crucial for determining sense of the word, makes VerbaLex suitable for disambiguation of polysemous and homonymous verbs. This paper tries to manifest this via investigation of valency frames of polysemous and homonymous verbs. Some related aspects of VerbaLex and its contents are discussed too.

1 Introduction

Polysemy and homonymy are phenomena which cause many problems in natural language processing. It is easier to solve the latter as was shown by [1] and [2]: accuracy about 95% can be attained in disambiguation of homonyms.

Homonymy is accidental phenomenon and that is why two homonymous words usually differ a lot in their behaviour and contexts. E.g. two homonymous Czech verbs *sladit*. Their meanings are a. *to sweeten* and b. *to coordinate*. Obviously, their usual contexts differ a lot.

Polysemy is far harder task to deal with. Accuracy of a solution depends strongly on granularity of senses in a reference dictionary. In the case of a fine-grained sense distinction even human annotators may not agree each other on particular disambiguation.

Since supervised methods perform better than other approaches [3, p. 56] it is reasonable to set eyes on data sources with semantic annotations. VerbaLex falls into this class. In this article we will discuss its possible contribution to *verb sense disambiguation* (VSD).

2 VerbaLex

VerbaLex is valency lexicon of Czech verbs. Each verb lemma, together with a number of its sense, represents a *literal*. Synonymic literals are grouped into *synsets* – basic elements of both WordNet [4] and VerbaLex. Each synset in VerbaLex has list of *frames* valid for particular literals from the synset. VerbaLex

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 9–14, 2010.

nowadays contains about 6,300 verb synsets, 21,000 literals, 10,500 verb lemmas and 20,000 frames. For more detailed description of VerbaLex structure see [5]. Here we will describe only its most important component – *frame*.

2.1 VerbaLex Frame

A frame includes combination of semantical, syntactical and morphological information about context of a particular verb. The frame is represented by a list of *semantic roles* with important additional information. There are 29 roles such as AG (agens), LOC (location), PAT (patient) etc. Verbs themselves have label VERB in frames.

Additional information is

- obligation: if a role is obligatory or optional in the frame,
- semantic class: a set of possible words on a position of the role represented by WordNet literal and
- morphological and syntactical constraints of the role, e.g. direct or prepositional case, animality etc.

For more detailed description of additional information, again, see [5]. Example of valency frame for verb *mačkat* follows:

$$\mathsf{AG}^{kdo1}_{person:1} + \mathsf{VERB} + \mathsf{OBJ}^{co4}_{object:1} + (\mathsf{PART}^{v\ čem6}_{hand:1}).$$

Optional roles are surrounded by parentheses and additional information is in superscripts and subscripts. The frame is formal representation of usual verb behaviour. A realisation of the frame may be e.g. *Honza mačkal bankovky v ruce.* (*John was pressing bank-notes in his hand.*). All constraints are met: agens *Honza* is in nominative and is animate (*kdo*1, i.e. *who* in animate nominative), object *bankovky* is in accusative and is inanimate (*co*4, i.e. *what* in inanimate accusative) etc.

Frames are core of VerbaLex. They describe the majority of possible contexts of verbs which might be used in disambiguation of polysemous and homonymous verbs.

We assume that each meaning of a verb has its own specific context. Since local contexts are represented by frames in VerbaLex, we can evaluate disambiguational potential of VerbaLex by checking uniqueness of these frames.

3 Frames of Polysemous and Homonymous Verbs

3.1 VerbaLex as Python Data Structure

VerbaLex comes in two formats: XML and text format. Our simple procedure goes through VerbaLex and converts it into a dictionary of frames. Keys of the dictionary are lemmas and its value is a list of couples: (number of a meaning of lemma, a list of valid frames for an appropriate literal). Frames are lists of semantic roles and a semantic role bears mentioned additional information.

3.2 Comparison of Frames

After the converting, another procedure compares all frames of all pairs of polysemous and homonymous verbs (lemmas). Thanks to described structure of the dictionary this step is quite straightforward.
The number of all possible pairs is expressed by the following formula:

$$\sum_{v \in V} \sum_{i,j \in S_v, i<j} min\left(|\mathrm{F}_v^i|, |\mathrm{F}_v^j|\right).$$

Structure of the formula corresponds (to a certain extent) to steps of the latter procedure: for each homonymous or polysemous verb (lemma) v in VerbaLex (V) and for each pair of meanings i and j from a set of meanings of v (S_v), the procedure compares all combinations of pairs of frames. F_v^i represents a set of frames for a given lemma v and its meaning i. Cardinality of this set is $|\mathrm{F}_v^i|$.

The formula expresses the highest number of possible identical frames between all pairs of meanings. In the case of VerbaLex it equals to 190,795. The second procedure looks for frames which are identical.

4 Results

The number of identical frames depends on several criteria. At first we looked for absolute identities. I.e. the case when two frames are identical in all information they bear: semantic roles and all additional information. It yielded 891 pairs of literals with at least one identical frame.

Then we checked less strict identity: two frames were considered as identical if they consisted of same obligatory semantic roles (whereas the previous identity takes into account obligatory as well as optional semantic roles) together with appropriate additional information. In that case there were 2,203 identical frames.

The third option was to compare only names of semantic roles, i.e. to omit additional information. This option yielded 3,343 frames.

Since substantial number of matches corresponded to perfective and imperfective variants of verbs which share same frames, we manually removed all imperfective variants and obtained 605 pairs of literals as the fourth option.

Other synonymic variants of verbs (two forms of infinitive – *pomoci* and *pomoct*) also share frames so number of really identical frames in Verbalex is even smaller.

Results are summarized in Table 1 on the following page.

4.1 Examination of Identical Frames

Results shown above are very promising. Out of 21,000 literals, only 605 share a frame. Moreover, if we take a look at concrete pairs of literals with identical frames we will discover that utter majority of them are rather annotation inconsistencies than real identities between frames. We can classify these identities into 3 groups.

Table 1. Number of identical frames according to various criteria

quantity	‰	**identity criterion**
891	4.67	absolute identity
2,203	11.55	identity only for obligatory semantic roles
3,343	17.52	identity of semantic role names
605	3.17	absolute identity without perfective and imperfective variants
190,795	10^3	all possible identities

Invalid Verb in Subsynset Since frames are usually not valid for all literals in a synset, they are assigned to subsynsets. A lexicographer must decide for which verbs in a synset a frame holds and then to create an appropriate subsynset for the frame.

Frame shared between literal *mrkat*:2 (*to wink*) and *mrkat*:3 (*to watch something with interest intermittently*) is

$$\mathsf{AG}^{kdo1}_{person:1} + \mathsf{VERB} + \mathsf{PAT}^{na\ koho4}_{person:1}.$$

The subsynset which contains the literal *mrkat*:2 contains also literals *mrknout*:2 and *zamrkat*:1. It is all right since the two literals are just perfective variants of *mrkat*:2. The problem is with the subsynset which contains the literal *mrkat*:3. It also contains literals *mrknout*:3 (perfective variant of *mrkat*:3), *pokukovat*:2 and *pomrkávat*:1. In the case of this subsynset, the first two literals (including *mrkat*:3) are not valid for this frame and should be removed from the subsynset.

Insufficient Distinction by Semantic Class Some verbs whose senses are distinguished very finely differ in details. If the only distinction between them is on lexical level and there is not suitable semantic class in WordNet, they can not be distinguished by valency frame itself. Literals *hořet*:1 and *hořet*:4 may serve as example. Their share frame

$$\mathsf{SUBS}^{co1}_{substance:1} + \mathsf{VERB}.$$

The first literal stands for *undergo combustion* and the second for *glow*. It is hard to imagine some substance which is burning and at the same time is not glowing. If we wanted to distinguish these cases we would need semantic classes for substances which are glowing whilst burning and for substances which can burn without emitting any light.

There are also other annotation errors in frames, especially in additional information: wrong animality, prepositional case etc. These mistakes should also be corrected.

Too Fine-Grained Distinction between Verb Senses In some cases there are very fine differences between senses which can not be distinguished by frames.

Frame

$$\mathsf{AG}^{kdo1}_{person:1} + \mathsf{VERB} + \mathsf{ART}^{co4}_{artifact:1} + (\mathsf{SUBS}^{čím7}_{material:1}).$$

is shared between literals *pokreslit*:1 and *pokreslit*:2. Both verbs have meaning *to deface*. The former meaning is rather neutral – *to cover with paintings*, whereas the latter is more negative – *to deface and to depreciate something by that*. This distinction can not be expressed in VerbaLex using only frames and additional information.

5 Conclusion

The second and the third group point to general problem of fine-grained word senses in natural language processing. We are able to distinguish tens of senses per word but then we are not able to distinguish between them in real applications automatically: the more senses we have the worse are results of word sense disambiguation. The question is whether we need to have all these fine-grained senses in our dictionaries at all.

Nevertheless, the experiment proved that VerbaLex could be very useful for *verb sense disambiguation* of polysemous and homonymous verbs. If we recognised a frame of a polysemous verb in a sentence, we would attain high precision in VSD. And since many synsets in VerbaLex are linked to appropriate synsets in English WordNet, this VSD could be used directly in machine translation from Czech to English.

6 Future Work

Our goal is to check all pairs of verb senses with identical frames manually and, if possible, to correct annotation errors. The procedure should be tuned to be fast enough and not to enable importation of new errors by an annotator.

The second goal is to check all frames in VerbaLex for soundness. The main endeavour is to find out whether all frames in VerbaLex are well-founded using a Czech corpus. Then we plan to discover which frames are useless or, on the contrary, which frames must be added to increase coverage of VerbaLex [6].

Both should improve consistency and quality of language data in VerbaLex.

Acknowledgements

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536. I want also to thank people involved in building VerbaLex for their respectable work on the advanced data source.

References

1. Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (Cambridge, MA). 189–196. 1995.

2. Stevenson, M., Wilks, Y. The interaction of knowledge sources in word sense disambiguation. Computational Linguistics 27, 3, 321—349. 2001.
3. Navigli, R. *Word Sense Disambiguation: A Survey*. ACM Computing Surveys, 41 (2), 2009, pp. 1–69.
4. Fellbaum, C. *WordNet: An electronic lexical database*. The MIT Press, 1998.
5. Hlaváčková, D., Horák, A. *VerbaLex – New Comprehensive Lexicon of Verb Valencies for Czech*. In Computer Treatment of Slavic and East European Languages. Bratislava, Slovakia: Slovenský národný korpus, 2006. pp. 107–115.
6. Jakubíček, M., Kovář, V., Horák, A. *Measuring Coverage of a Valency Lexicon using Full Syntactic Analysis*. In Sojka, P., Horák, A. (Eds.): RASLAN 2009: Recent Advances in Slavonic Natural Language Processing. Brno: Masaryk University, 2009. pp. 75–79.

Editing of VerbaLex

Dana Hlaváčková and Vašek Němčík

NLP Laboratory, Faculty of Informatics
Masaryk University, Brno, Czech Republic
xnemcik@fi.muni.cz, hlavack@fi.muni.cz

Abstract. In this article we point out some problems in editing of the valency database VerbaLex. Editing is mostly manual and time-consuming work, requiring experienced annotators. We propose a solution in the form of a new editing interface, which can be used by new annotators without making unnecessary mistakes. Further, we also mention creating new and modifying existing tools (WordNet Assistant), which can accelerate the work on VerbaLex and eliminate errors in the editing to a minimum.

1 Introduction

VerbaLex represents an extensive database of Czech verbs valency frames. The database is currently being created at the Natural Language Processing Centre at the Faculty of Informatics, Masaryk University. VerbaLex is a work which lies between linguistics and the sphere of natural language processing. This is a special type of synchronous dictionary built with the help of computer tools and the use of electronic data sources.

2 VerbaLex

The organization of lexical data in VerbaLex is derived from the WordNet [1] structure. It has a form of synsets arranged in the hierarchy of word meanings (hyper-hyponymic relations). For this reason, the headwords in VerbaLex are formed by lemmata in synonymic relations followed by their sense numbers (standard Princeton WordNet, henceforth PWN, notation). For each of the synsets, a definition describing the meaning of the verbs is added. At present there are also links between Czech and English equivalent synsets in the PWN [2] (and similar in other languages), which are ensured by the so-called interlingual index.

Information about verbs is captured in VerbaLex in the following way. The head of each entry is constituted by synonyms with the numbers of their verb meanings. The synset notation also bears information about existing aspect pairs, and the abbreviation of verbal aspect (*pf*, *impf*, *biasp*). The valency frames follow with the description of valency and semantic roles for each slot and examples of usage. The entire entry is accompanied by various attributes of verbs. Item *use* shows the use of verbs in natural context – *primary*, *figurative* or *idiomatic*. The

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 15–20, 2010.

type of reflexivity is marked for reflexive verbs. The last item refers to a system of semantic classes. This numbering is based on semantic classes by B. Levin [3] and was adopted from the project VerbNet [4], formulating a very detailed segmentation of English verb meanings into 395 classes. For our purposes it was reduced to 100 classes of Czech verbs. VerbaLex is available in TXT, PDF, XML and HTML formats.

3 Creating and Editing of VerbaLex

VerbaLex has been under development since 2005 and currently contains 10,482 verbs and 19,556 valency frames. 15 annotators and 6 technical support staff members have participated on the creation and editing of VerbaLex.

Originally, the valency database in its basic form could be edited using an interactive tool verbalex.sh, which is based on a well-configurable multi-platform editor GVIM. It allows easy insertion of language data in plain text. The advantage of the editor is syntax highlighting, which in our case, streamlines editing of the entire text and provides immediate control of the accuracy of recorded data (incorrect text is highlighted in color). The editor was specially adapted for creating VerbaLex and offers a variety of other ways to speed up and facilitate the work with a large database. The default format of valency frames was offered to the annotators and their task was to fill it with the actual data.

Creating a database of verb valencies represents a large amount of manual work which is commonly associated with unintentional errors. GVIM editor has been modified to indicate the maximum number of procedural errors. Besides the aforementioned coloring, an automatic check of formal errors in the file processed was added to the editor. It allows each user to individually review the selected section, edit and correct the majority of errors both in the format of an entry, and in the logical structure of the valency frame. Further errors are highlighted during the export of VerbaLex to the XML and HTML format. With the new version of the editor (VIM 7.0), automatic control of Czech spelling was added, which greatly facilitates the correction of errors and typos in the text parts of the database (e.g. definitions and examples).

4 Present Situation and Difficulties in Editing

Currently, editing VerbaLex is possible without having to open the large text file containing the database. After the annotator enters the verb, separate windows are opened with only the synsets containing entered verb. In this way we can also add new verbs to the database (when opening a blank window).

In spite of all efforts to ensure the correct editing of VerbaLex, it is still manual work with text files, which are prone to random errors that are difficult to detect. This type of errors cannot be found by the control in GVIM, because it was not designed to detect this type of inconsistencies. The errors are often discovered

only in the final version of VerbaLex, in HTML browser. The most common errors are:

- missing parts of text, such as tag for the verb in valency frame (VERB);
- accidental deletion of parts of text;
- bad copy of parts of the text.

The number of errors increases with each editing by inexperienced and untrained annotators. Moreover, we need to spend lots of time to explain the way of notation in the GVIM editor with formal errors control to new annotators.

For existing and new annotators is is also difficult to add new verbs. It is necessary to observe many rules in terms of both content and in terms of formal notation. Adding new verbs consists of several steps, which are an opportunity for inclusion of new errors. This process is time-consuming work and is performed manually.

For the verb, which is not yet listed in the database, it is necessary to, above all:

- find appropriate synonyms and build synsets;
- write a definition that covers all the meanings of the verbs in synset;
- create a valency frame indicating semantic roles and examples of use (frames are often not valid for the whole synset, but for individual verbs only, that are arranged in the so-called subsynset);
- add any other information for individual verbs (aspect, reflexivity, the type of use);
- include synset to the semantic classes;
- find an English equivalent synset in PWN and verify that it is not already linked to another Czech synset.

Our goal is to simplify the work and shorten the time required to add new verbs. One possibility is the use and adaptation of an existing tool, the WordNet Assistant, for searching English equivalents in PWN. Without its use, the annotators have to:

- translate Czech verb into English;
- find the English equivalent in the PWN (usually occurs in several synsets);
- choose the correct English synset;
- determine whether it is not linked to some other Czech synset (manual search in text version of VerbaLex).

This part of work has been simplified by creating a new version of the WordNet Assistant. Its use eliminates translation errors in selecting the right verbs and equivalents in PWN.

5 WordNet Assistant

Extending a valency lexicon such as VerbaLex is a complex task that cannot be performed automatically reliably enough. For humans, however, challenging and interesting as it is, larger-scale lexicographic work is rather tedious and humdrum. Human work may therefore often be rather slow and prone to mistakes.

This led us to implement a web-based application, the WordNet Assistant (henceforth WNA). It aims at assisting the human lexicographer at making various decisions common when adding new words or whole synsets to VerbaLex and thus speeds up the editing process. Moreover, it helps discovering and preventing certain types of inconsistencies.

The original form of WNA was designed to assist at adding whole new synsets by suggesting their probable English counterparts in the PWN. More information about it can be found in [5].

The currently relevant form of WNA concerns individual words and it gives the lexicographers the opportunity to view them in a broader context. Given a Czech word (and optionally its part-of-speech), it presents a list of relevant English synsets in PWN, supplemented by references to existing VerbaLex synsets.

The computation proceeds in two main steps:

- Czech-English dictionary lookup,
- PWN lookup using the DEB server.

The dictionary lookup is carried out using the GNU/FDL English-Czech Dictionary compiled at the Technical University in Plzeň [6]. In principle, any Czech-English dictionary available can be used in this step. The translations found in the dictionary are presented, and only those selected by the user are used in the subsequent step. This manual step accounts for disambiguation problems and possible noise in dictionary data.

The English translations resulting from the previous steps are passed as a query to PWN, with use of the DEB server [7]. A list of synsets containing the individual translations as literals is presented, synsets containing translations of more input words ranked higher on the list. The synsets on the list represent the range of relevant senses to be straightforwardly found in WordNet.

The synsets on the list are not accompanied only by their definition and examples, further, information about their respective counterparts in VerbaLex is included. This information is of great help when looking for inconsistencies, such as VerbaLex synsets that need to be split or merged. It also facilitates locating VerbaLex synsets the given word may be added to, or related senses that haven't been covered in VerbaLex yet.

The functionality and information provided by the application will be adapted based on the current needs and experience of the lexicographers.

6 New Editing Interface for VerbaLex

The question is, how to avoid errors in editing the database and speed up the work when adding new verbs. One possibility is to create a new user interface for editing of VerbaLex, which will be sufficiently clear and user-friendly for new and inexperienced annotators. User interface should be easily accessible and manageable (e.g. through a web interface), and should meet certain basic requirements that will ensure faster and more efficient editing of the database.

The requirements are:

1. clear graphic design of the interface with a simple navigation for users;
2. clearly separate the attributes of verbs and attributes of synsets;
3. add option to edit the individual verbs (with all their attributes), not all synsets only;
4. reduce manual text input to a minimum, leave this option only for definitions and examples (if possible, implement spell checking for these parts);
5. inputting attributes of verbs and synsets allowed only by selecting default options;
6. separate the left-side and right-side parts of valency frames;
7. check blank parts of the interface form;
8. preview of the finished entry;
9. record of the annotator's name and date of editing;
10. setting different levels of access rights (e.g. view, edit, add new entry).

This is a summary of the basic requirements, which should minimize the random formal errors that occur when handling a text file. The interface allows entering free text only in case of synset definitions and usage examples for valency frames, other items will be selected from a fixed menu of options. Transparent record of the names and dates allow us to check the work of annotators. Distinction of the access rights prevents accidental and unauthorized access to the database. When designing the actual interface we probably design other options that will ensure a simple and clear VerbaLex editing.

7 Conclusions and Further Work

A new editing interface should help to eliminate formal errors resulting from inattentive editing and accelerate the work on VerbaLex. With regards to this, it will also be necessary to change the existing XML format and a web interface, which could be supplemented by more options for searching information in VerbaLex. The WNA tool now allows efficient search of English equivalents in PWN and simplifies the addition of new verbs. The question remains, how to find and correct various inconsistencies in the database contents. It is often impossible to identify them automatically and they can corrected only by manual browsing of VerbaLex. One of our other tasks is identification of irregularities in the database content and finding ways of remove them efficiently.

Acknowledgements

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536.

References

1. Fellbaum, C.: Wordnet. An electronic lexical database (1998).
2. Fellbaum, C.: English verbs as a semantic net. In: Five papers on WordNet, Princeton University (1990) Technical Report 43, Cognitive Science Laboratory.
3. Levin, B.: English Verb Classes and Alternations: A Preliminary Investigation. The University of Chicago Press, Chicago (1993).
4. Palmer, M., Rosenzweig, J., Dang, H.T., Kipper, K.: Investigating regular sense extensions based on intersective levin classes. In: Coling/ACL-98, 36th Association of Computational Linguistics Conference, Montreal (1998) 293–300.
5. Němčík, V., Pala, K., Hlaváčková, D.: Semi-automatic linking of new Czech synsets using Princeton Wordnet. In: Intelligent Information Systems XVI, Proceedings of the International IIS '08 Conference, Warszawa, Academic Publishing House EXIT (2008) 369–374.
6. Svoboda, M.: GNU/FDL English-Czech dictionary (2008) `http://slovnik.zcu.cz/`.
7. Horák, A., Pala, K., Rambousek, A., Povolný, M.: DEBVisDic – First Version of New Client-Server Wordnet Browsing and Editing Tool. In: Proceedings of the Third International WordNet Conference – GWC 2006, Brno, Czech Republic, Masaryk University (2005) 325–328.

Morphological Analysis of Tajik
Notes and Preliminary Results

Gulshan Dovudov and Vít Baisa

Natural Language Processing Centre, Faculty of Informatics
Masaryk University, Brno, Czech Republic
387176@mail.muni.cz, xbaisa@fi.muni.cz

Abstract. In this article we describe state of art of morphological analysis of Tajik language. At first we comment retrieval of prefixes and postfixes. Then we introduce an algorithm for semi-automatic morphological analysis of one-root Tajik words. The algorithm works with a database of roots, prefixes and suffixes and in the case a new root or a new affix is found the algorithm adds it into the database on the basis of manual analysis.

Key words: Tajik language, morphological analysis, prefix, root, postfix, suffix, affix

1 Introduction

The Tajik language belongs to the Iranian group of languages which is a part of the extensive Indo-European language family. With its grammatical system, Tajik language belongs to the group of languages of analytic type. A rich system of inflectional case forms existed in the ancient Iranian languages but it is fully lost in present Tajik. Case forms in Tajik are expressed by purely syntactical means: prepositional and postpositional construction, *izafet* combination and word order. A category of gender is also almost lost despite it existed in the ancient Iranian languages too. In Tajik, only verbs have a developed system of syntactical and analytical forms.

Generally, every Tajik word can be segmented into three parts – *morphemes*: *prefix*, *root* (the lexical kernel of a word bearing its basic semantic value) and *postfix* (*suffix* + *ending*). That is why they can be expressed as one of four models: R, Pr $\oplus$ R, Pr $\oplus$ R $\oplus$ Ps and R $\oplus$ Ps (Root, PRefix and PoStfix). Table 1 shows some examples of Tajik words segmented into morphemes.

2 Definitions

Let us mention some definitions applicable to Tajik language:
Root – the main part of a word. Root is the mandatory part of every word.
Affix – an auxiliary part of a word added to the root which serves to word formation and expressing of grammatical meanings. Affixes form words only in conjunction with roots. Affixes alone do not bear any lexical meaning.

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 21–27, 2010.

Table 1. Examples of Tajik words segmented into morphemes

Structure	Tajik word	In latin	Translation
R	китоб	kitob	book
R	кор	kor	work
R	зиёд	ziyod	many
R	ист	ist	stand
R ⊕ Ps	китоб-ча	kitob-cha	litle book
R ⊕ Ps	кор-гар	kor-gar	worker
R ⊕ Ps	соя-бон	soya-bon	tilt
R ⊕ Ps	шарм-гин	sharm-gin	timid
Pr ⊕ R	но-умед	no-umed	despair
Pr ⊕ R	бар-зиёд	bar-ziyod	excessive
Pr ⊕ R	на-рав	na-rav	don't go
Pr ⊕ R ⊕ Ps	бар-омад-ан	bar-omad-an	to ascend
Pr ⊕ R ⊕ Ps	(на-ме)-рав-и	(na-me)-rav-i	you do not go
Pr ⊕ R ⊕ Ps	(на-ме-фур)-омад-ам	(na-me-fur)-omad-am	I do not descend

Prefix – a morpheme standing before the root and changing its lexical or grammatical meaning. Prefixes are divided into two groups: simple and compound. Compound prefixes (disyllabic and trisyllabic) are formed by concatenating of appropriate number of simple prefixes.
Postfix – a part of a word which follows directly a root, consisting of suffixes and endings. Postfixes as well as prefixes are divided into two groups, but a compound postfix can consist of 2–8 simple suffixes.
Suffix – a kind of affix morpheme which follows a root and comes before an ending.
Base – a part of a word that remains after the cutting-off ending. A base may be only root or root with affixes.

3 Database of Morphemes

3.1 Postfixes

In this paper, we assume that every input word is correct. I.e. there are no errors in spelling. The procedure of morphological analysis is based on previously prepared fixed database of morphemes – roots, prefixes and postfixes.

Morphological analysis of a word equals to segmenting that word into three mentioned components.

Database of postfixes of Tajik literary language was expanded step by step based on iterative processing of representative texts (see Section 6). As a result, database of 2,533 suffixes with their frequencies of occurrence was made.

Table 2 shows the frequency of postfixes of different level of complexity. The level represents number of simple postfixes in compound postfix. 0 level of complexity means that there is no postfix in a word.

Table 2. List of postfixes with frequencies

L	Count	Frequency
0	0	46.89650
1	113	39.25153
2	755	11.12421
3	1,017	2.35906
4	540	0.35571
5	86	0.01142
6	17	0.00129
7	3	0.00019
8	2	0.00006

3.2 Prefixes

Database of prefixes was created combinatorially and elaborated by statistical method.

We have complete list of simple (one-syllable) prefixes at our disposal: ба (ba-), бар (bar-), бе (be-), би (bi-), бо (bo-), боз (boz-), бу (bu-), во (vo-), дар (dar-), ма (ma-), ме (me-), на (na-), но (no-), то (to-), фар (far-), фур (fur-), ҳам (ham-), ҳаме (hame-) and ҳар (har-).

These 19 prefixes represent all simple prefixes. Since any compound prefix may be created as a concatenation of two or three simple prefixes we can generate all double and triple prefixes by permutations. There are 342 (19*18) double and 5,814 (19*18*17) triple possible prefixes. It is obvious that simple prefixes may not repeat in compound prefixes.

These hypothetical prefixes were checked semi-automatically and the result was list of 19 simple, 39 double and 8 triple real prefixes.

Table 3 provides a list of all currently known prefixes ordered by their frequency. Frequencies and counts for both prefixes and suffixes were derived from representative texts (see Section 6).

3.3 Coverage of the Database

At the moment it is difficult to even estimate the coverage of our database of morphemes.

Processing of about 1,700,000 words yielded 66 prefixes, 26,479 roots and 2,533 postfixes. After processing of other texts (about 1,140,000 words) we obtained only 2 new prefixes, 4,443 new roots and 360 new postfixes. It is about 4.5% of new prefixes, about 16.77% of new roots and 14.21% of new postfixes.

Since these new morphemes have very low frequency we cam assume that the coverage is considerably high.

Table 3. List of prefixes with frequencies

#	Prefix	Freq.	#	Prefix	Freq.	#	Prefix	Freq.
1	ме (me)	48.356	23	барна (barna)	0.048	45	бознаме (bozname)	0.003
2	на (na)	11.421	24	воме (vome)	0.048	46	меби (mebi)	0.002
3	бе (be)	7.113	25	бозме (bozme)	0.031	47	нафур (nafur)	0.002
4	ҳам(ham)	6.913	26	мефар (mefar)	0.029	48	вонаме (voname)	0.002
5	бар (bar)	6.369	27	наби (nabi)	0.024	49	намефур (namefur)	0.002
6	наме (name)	5.033	28	барнаме (barname)	0.021	50	ҳаме (hame)	0.002
7	но (no)	3.568	29	намедар (namedar)	0.020	51	бебоз (beboz)	0.001
8	бо (bo)	2.639	30	дарме (darme)	0.015	52	бифар (bifar)	0.001
9	би (bi)	2.616	31	ноба (noba)	0.013	53	бубар (bubar)	0.001
10	ҳар (har)	1.128	32	бино (bino)	0.009	54	мена (mena)	0.001
11	ба (ba)	1.055	33	баҳам (baham)	0.008	55	ноби (nobi)	0.001
12	дар (dar)	0.681	34	надар (nadar)	0.008	56	нодар (nodar)	0.001
13	боз (boz)	0.675	35	бано (bano)	0.006	57	ноҳам (noham)	0.001
14	ма (ma)	0.386	36	дарбар (darbar)	0.006	58	ҳамебар (hamebar)	0.001
15	во (vo)	0.377	37	дарна (darna)	0.006	59	намефар (namefar)	0.001
16	мебар (mebar)	0.362	38	барҳам (barham)	0.005	60	фар (far)	0.001
17	барме (barme)	0.316	39	бозна (bozna)	0.005	61	беба (beba)	0.001
18	бу (bu)	0.267	40	вона (vona)	0.004	62	бозма (bozma)	<0.001
19	то (to)	0.124	41	мефур (mefur)	0.004	63	вома (voma)	<0.001
20	медар (medar)	0.101	42	нафар (nafar)	0.004	64	дарма (darma)	<0.001
21	набар (nabar)	0.073	43	дарнаме (darname)	0.004	65	фур (fur)	<0.001
22	Намебар (namebar)	0.058	44	дарҳам (darham)	0.003	66	барма (barma)	<0.001

4 Semi-Automatic Morphological Analysis

Quality of semi-automatic morphological analysis of a word strongly depends on the database of morphemes. An output of the analysis is either a segmentation of a word into three parts (Pr, R and Ps), or information that the word can not be segmented into known morphemes. It is quite clear that a negative result is a consequence of incompleteness of our database.

For this reason it seems natural to expand the database by adding new morphemes manually identified by an expert during morphological analysis, see Section 5.

The algorithm for semi-automatic morphological analysis for Tajik words is depicted in the form of flowchart on Figure 1 on the next page.

We have all Tajik words consisting of one or two letters in our database and therefore the analysis will process only words with strictly more than two characters. If the analyser gets one- or two-character word it immediately outputs result, i.e. root.

Morphological analysis of a word consists of the following steps. Block 1 represents recognition of a prefix. Since Tajik prefixes contain at least two letters, we pick two letters from the beginning of input. Then we select all prefixes from database which start with these two letters.

If none of these selected prefixes is contained in the word it is natural to assume that the word begins with the root. If the prefix is identified it is removed from the word, and the remaining fragment of the word is analyzed in block 2.

The process in block 2 is similar to the previous block. If at least one root is found then it is removed from the word and, again, the remaining fragment goes to block 3. Here it is compared with postfixes from the database. If at least

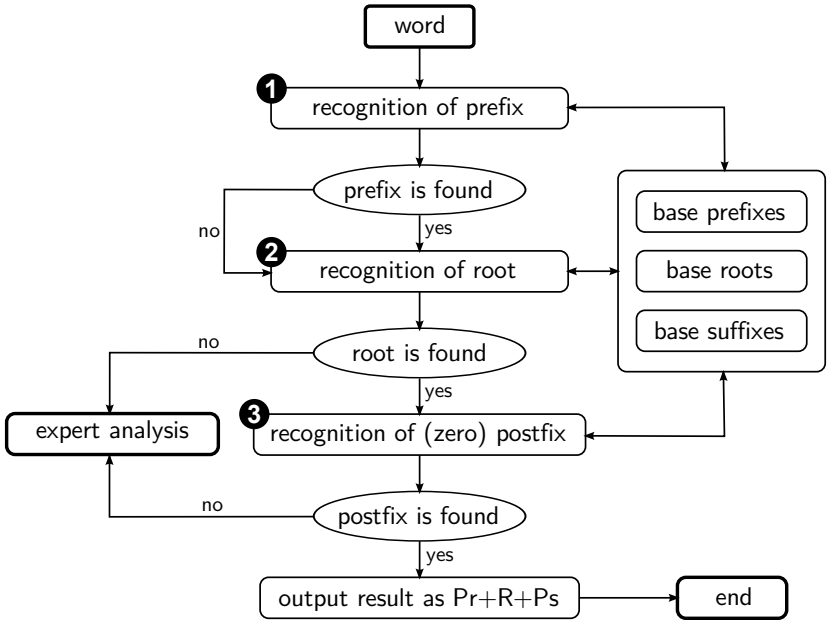


Fig. 1. Algorithm for morphological analysis

one postfix is found the process is completed successfully and the input word is represented as a concatenation of three morphemes – prefix, root and postfix.

It is quite clear that an analysis of some words may result in only root or root with either prefix or postfix.

It is also obvious that morphological analysis of some words can not find any possible segmentation into morphemes and since we expect correctly spelled word, this phenomenon is caused by the fact that the database do not contain corresponding morphemes. In such cases the word is sent to manual analysis.

5 Linguistic Analysis

If morphological analysis of a word fails, the word has to be analysed manually. A language expert segments the word into morphemes. The algorithm then determines whether any of these morphemes are already in the database. If not, the new morphemes are added.

6 Data Resources Description

Representative sample of about 8,000 pages (about 4,000,000 words) was used for text processing described above. Texts were taken from literary works, newspaper articles and from professional literature in Tajik language. For more details, see Table 4 on the following page.

Table 4. Texts used for text processing

No.	Author	Document	Pages
1	Abu Ali Ibn Sino	AL-Konun	200
2	Abulkosim Firdavsi	Shohnoma	200
3	Sadriddin Ayni	Yoddoshtho	280
		Yatim	220
		Kahramoni khalki tojik – Temurmalik	150
4	Bobojon Gafurov	Tojikon	200
5	Sotim Ulugzoda	Piri hakimoni mashrikzamin	150
6	Nazirjon Tursunov	Ta'rikhi tojikon	400
7	Muhammadjon Shakuri	Panturkizm va sarnavishti khalki tojik	346
		Khuroson ast injo	360
		Sadri Bukhoro	187
8	F. Muhammadiev	Kulliyot	100
9	L. Sherali	Namunai ash'or	300
10	Jalol Ikromi	Asarhoi muntakhab	100
11	Abdumalik Bahori	Bozgasht	100
		Sohili Murod	100
12	Rahim Jalil	Odamoni Jovid	100
13	M.Ganiev	MS'Word	50
14	Hakim Rahimi	Oila va oiladori	150
		Farhangi zaboni tojiki	150
15	newspapers	Jumhuriyat	270
		Sugd	280
		Sadoi mardum	200
		Charkhi gardun	400
		Lochin	192
		Mash'al	181
		Nohid	161
		Salomat Boshed	1,040
		Sukhani Khalk	1,309
		Khirman	100

7 Future Work

Our goal is to extend the database of morphemes by processing other literary texts gathered from electronic books, newspapers, internet etc.

We will also put all available documents together and make a corpus of Tajik with more than 5,000,000 millions of tokens.

Extensive work on the morphological analyser should also lead to development of a spell checker, POS tagger and an algorithm for morphological disambiguation.

With these tools we will be able to annotate data in the corpus automatically.

All these goals should hopefully end with high-quality data sources of Tajik language.

8 Tajik Language Processing – State of Art

There are few works connected with Tajik language processing. Besides localisation of some software and creating Tajik keyboard layout [9] it is necessary to mention Russian-Tajik and Tajik-Russian dictionary (Usmanov and Soliev) and text-to-speech synthesizer [10,11] based on syllables.

Acknowledgements

This work has been partly supported by Erasmus Mundus Action II lot 9: Partnerships with Third Country higher education institutions and scholarships for mobility, and by the Ministry of Education of CR within the Center of basic research LC536.

References

1. Rastorgueva, V. S. *A brief sketch of the grammar of the Tajik language (supplement to the Tajik-Russian Dictionary)*. Moscow: State Publishing House of Foreign and National Dictionaries, 1954, pp. 529–570.
2. Buzurgzoda, L., Niyazmuhammedov, B. *Grammar of the Tajik language. Part 1: Phonetics and Morphology*. Stalinabad. 1944, 112 p.
3. Active Tajik literary language. Volume 1: Lexicology, phonetics and morphology. Dushanbe: Irfon, 1973. pp. 452.
4. Rustamov, S. *Derivation of nouns in the modern Tajik literature language*. Dushanbe, 1972, pp. 90.
5. Amonova, F. R. *Noun affixal derivation in modern Persian and Tajik languages*. Dushanbe, 1982, pp. 55.
6. Usmanov, Z. D., Dovudov, G. M. *On forming the prefix base to the literary Tajik.* Reports of the Academy of Sciences of the Republic of Tajikistan, no. 6, 2009, pp. 431–436.
7. Usmanov, Z. D., Soliev, O. M., Dovudov, G. M. *On a set of postfixes of Tajik literature language*. Reports of the Academy of Sciences of the Republic of Tajikistan, no. 2. 2010, pp. 99–103.
8. Usmanov, Z. D., Dovudov, G. M. *On statistical regularities of tajik morpheme basis*. Reports of the Academy of Sciences of the Republic of Tajikistan, no. 3. 2010, pp. 188–191.
9. Soliev, O. M. *Mathematical model of optimal keyboard layout and its applications*. Ph.D. thesis. `http://www.mitas.tj/dissov/kandidat/avtoreferat/o_soliev.pdf`
10. Khudoiberdiev, K. A. *Tajik Text-to-Speech Synthesizer*. Ph.D. thesis. `http://www.tajik-tts.narod.ru/`
11. Khudoiberdiev, K. A. *Complex of Program Synthesis Tajik* `http://www.mitas.tj/dissov/kandidat/avtoreferat/khudoiberdiev.pdf`

Part II

Text Corpora and Annotation

Legal Terms and Word Sketches
A Case Study

Eva Mráková and Karel Pala

Natural Language Processing Centre, Faculty of Informatics
Masaryk University, Brno, Czech Republic

Abstract. In this paper we describe an approach to the semiautomatic identification of legal terms in Czech texts. Our general goal is to offer supplementary tools for building dictionary of Czech law terms.
At first we used the VaDis partial parser for recognition of the complex nominal constructions in a legal text – the current version of the Penal Code of the Czech Republic. Headwords of the recognized structures are usually relevant legal terms. Then we employed the Sketch Engine to find Word Sketches of these relevant terms in a large corpus of the standard Czech Czes, because corpora of legal Czech texts are not available yet. In spite of the fact that we used common texts we obtained very good candidates for legal terms as a result.
We also discuss relations between VerbaLex frames of the selected group of Czech verbs with financial meaning that occur in legal texts and Word Sketches found for some of these verbs. It appears that the combination of the valency frames and Word Sketches provides good candidates for the legal terms as well.
The paper is conceived as a case study in which we describe collocational behaviour of the selected Czech noun phrases and also some verbs belonging to the financial domain.

1 Introduction

Previous work focused on legal term recognition in Czech texts is described in [1] and [2]. While the first paper concerns especially legal terms in the form of noun groups, the second has dealt with legal verbs and their valency frames. Here we present possible enhancements of the methods mentioned in these papers and we also suggest exploitation of other tools suitable for building a legal term dictionary. Another approach is described in [3] which mainly relies on manual processing of the legal texts when finding the legal terms.

2 Recognition of the Complex Nominal Constructions

The current version of the Penal Code of Czech Republic containing approx. 36,000 word forms had served as a data source for experiments described in [1]. These experiments were optimized for high speed and thus the partial

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 31–39, 2010.

Table 1. Examples of Complex Nominal Groups

complex nominal group	English equivalent
pachatel trestného činu	who committed a criminal act
spáchání trestného činu	committing a criminal act
pokus trestného činu	an attempt to commit a criminal act
znaky trestného činu	attributes of the criminal act
způsob provedení činu	way of the committing a criminal act
dokonání trestného činu	completing a criminal act
účastník trestného činu	participant of the criminal act
trestnost pokusu trestného činu	punishability of the attempt
doba spáchání činu	time of the committing a criminal act
povaha spáchaného činu	nature of the committed criminal act
stupeň nebezpečnosti činu	degree of the dangerousness of criminal act

parser VaDis [4] used for syntactic analysis was transformed into Perl regular expressions for this purpose. The result of the experiment were base forms of noun groups sorted according to their frequency and these groups should appear as entries in the legal electronic dictionary, which is in preparation [3].

We used the same data source but in comparison with the above mentioned approach we have been working with the original Prolog version of VaDis and focused on more complex nominal constructions and their hierarchical structuring. The analysis of the data took several minutes and it was acceptable without any need for an optimization. The recognized noun phrases were not sorted according to their frequency but were clustered according to their headwords. The headwords of big clusters were often essential legal terms like *čin (act)*, *trest (punishment)*, *pachatel (offender)*, *zákon (law)*, *sazba (penalty)*, *vazba (detention)*, *soud (court)*, *předpis (regulation)*, *opatření (measure)*, *následek (consequence)*, *škoda (damage)*, etc. Of course, there were also clusters whose headword's legal meaning requires wider context, such as *odpovědnost (responsibility*, *rozkaz (command)*, *společnost (society)*, *stát (government)* etc.

In each cluster we inspected the more complex nominal constructions. Table 1 shows some of such constructions for the cluster headword *čin (act)*.

It can be observed that (parts of) the complex nominal constructions are good candidates for legal "subterms" in the context of the particular headword, for instance, when describing the dictionary entry *(trestný) čin (criminal act)* we should describe (or refer to) the contextual words *pachatel (offender)*, *spáchání (commitment)*, *pokus (attempt)*, etc.

3 Word Sketches

Word sketches [5] are one page summaries of a word's grammatical and collocational behaviour and they represent a really helpful tool for building terminological dictionaries (and not only them). At the same time Word Sketch Engine produces information how firmly the individual members of the collocations are tied together – this is indicated by the salience parameter [6].

Based on the grammatical analysis, the Sketch Engine also produces a distributional *thesaurus* for a language, in which words occurring in similar settings, sharing the same collocates, are put together, and *sketch differences*, which specify similarities and differences between near-synonyms. The system is implemented in C++ and Python and designed for use over the web.

3.1 Obtaining Legal Terms through WSE

At the moment, unfortunately, we do not have any corpus of the Czech legal texts at our disposal which could be used as a direct source for the Sketch Engine. Instead, we have used big corpora of Czech common texts: SYN2000 [7] containing about 110 million tokens and Czes [8] containing 1,191,157,014 tokens (compiled at the NLP Centre FI MU and completed in 2009). It consists mostly of the newspaper texts downloaded from the Web. It is annotated grammatically with lemmas and POS-tags.

Although we did not use law texts as the source, the results of the Sketch Engine using the Czech sketch grammar [6] are interesting and quite promising. We explored Word Sketch tables for the headwords of clusters described in the previous section and we obtained several hundreds relevant legal terms from them.

Moreover, words in some Word Sketch tables form natural groups of legal terms. For instance, the Sketch table gen_1[1] for the headword *čin (act)* contains a reasonable classification of criminal acts. It is shown in Table 2.

Table 2. Sketch table gen_1 for *čin*

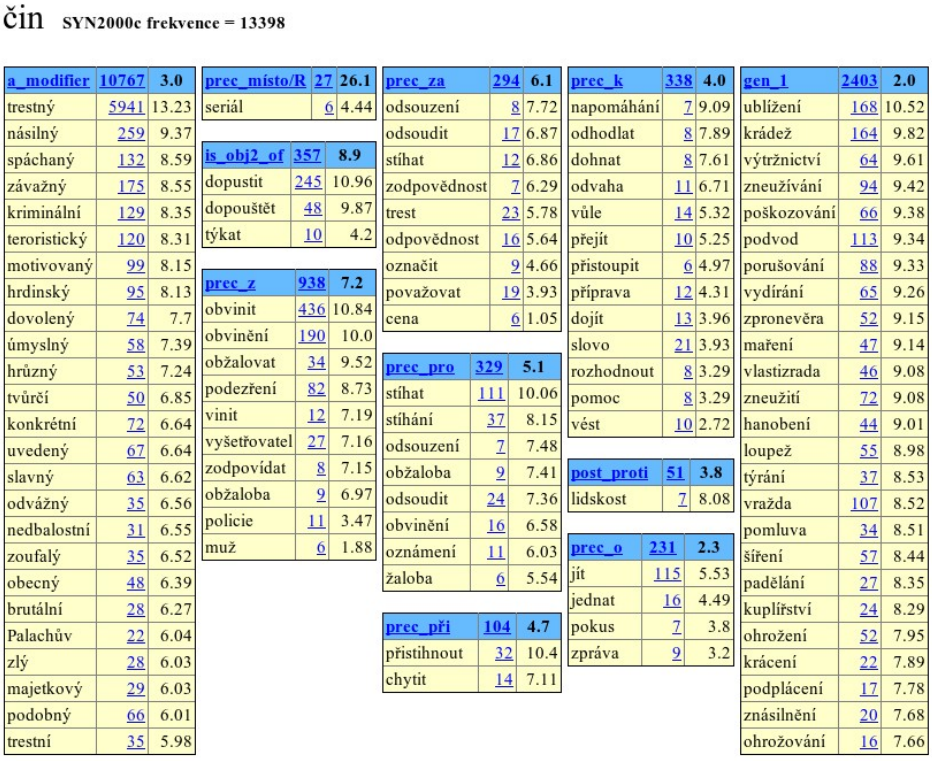
zpronevěra (defalcation)	padělání (forgery)
krádež (larceny)	hanobení (defamation)
poškozování (damaging)	podílnictví (shareholding)
loupež (robbery)	vražda (murder)
maření (obstruction)	zneužití (misaproppriation)
porušování (violation)	ohrožování (threatening)
zneužívání (misaproppriating)	zvýhodňování(privileging)
výtržnictví (disorderly behaviour)	ohrožení(emergency)
vydírání (extortion)	znásilnění (rape)
pomluva (slander)	podplácení (bribery)
zkrácení (reduction)	týrání (abuse)
zanedbání (negligence)	

Let us focuse on the headword *čin (act)* and its other Word Sketches. The word itself has a common meaning, not only the legal one. However, the Word Sketch Engine does not allow to process whole phrases like *trestný čin (criminal act)*, which is its "legal" specification. Despite of this, from general corpora consisting

[1] This is a very frequent collocation in Czech—it is a noun phrase consisting of the head noun and its dependent noun in genitive case, e. g. *pachatel trestného činu (the one who committed a criminal act – criminal, offender)*

of the common (e.g. newspaper) texts we obtained Word Sketch Tables with mainly legal terminology.

We have found almost 13,400 occurences of the *čin (act)* in SYN2000 and 88,500 ones in Czes. Corresponding Word Sketch tables contained more than one hundred words and more than two hundreds words respectively. A part of the Word Sketch Table of *čin (act)* obtained from SYN2000 is shown in Figure 1.



čin SYN2000c frekvence = 13398

a_modifier	10767	3.0
trestný	5941	13.23
násilný	259	9.37
spáchaný	132	8.59
závažný	175	8.55
kriminální	129	8.35
teroristický	120	8.31
motivovaný	99	8.15
hrdinský	95	8.13
dovolený	74	7.7
úmyslný	58	7.39
hrůzný	53	7.24
tvůrčí	50	6.85
konkrétní	72	6.64
uvedený	67	6.64
slavný	63	6.62
odvážný	35	6.56
nedbalostní	31	6.55
zoufalý	35	6.52
obecný	48	6.39
brutální	28	6.27
Palachův	22	6.04
zlý	28	6.03
majetkový	29	6.03
podobný	66	6.01
trestní	35	5.98

prec_místo/R	27	26.1
seriál	6	4.44

is_obj2_of	357	8.9
dopustit	245	10.96
dopouštět	48	9.87
týkat	10	4.2

prec_z	938	7.2
obvinit	436	10.84
obvinění	190	10.0
obžalovat	34	9.52
podezření	82	8.73
vinit	12	7.19
vyšetřovatel	27	7.16
zodpovídat	8	7.15
obžaloba	9	6.97
policie	11	3.47
muž	6	1.88

prec_za	294	6.1
odsouzení	8	7.72
odsoudit	17	6.87
stíhat	12	6.86
zodpovědnost	7	6.29
trest	23	5.78
odpovědnost	16	5.64
označit	9	4.66
považovat	19	3.93
cena	6	1.05

prec_pro	329	5.1
stíhat	111	10.06
stíhání	37	8.15
odsouzení	7	7.48
obžaloba	9	7.41
odsoudit	24	7.36
obvinění	16	6.58
oznámení	11	6.03
žaloba	6	5.54

prec_při	104	4.7
přistihnout	32	10.4
chytit	14	7.11

prec_k	338	4.0
napomáhání	7	9.09
odhodlat	8	7.89
dohnat	8	7.61
odvaha	11	6.71
vůle	14	5.32
přejít	10	5.25
přistoupit	6	4.97
příprava	12	4.31
dojít	13	3.96
slovo	21	3.93
rozhodnout	8	3.29
pomoc	8	3.29
vést	10	2.72

post_proti	51	3.8
lidskost	7	8.08

prec_o	231	2.3
jít	115	5.53
jednat	16	4.49
pokus	7	3.8
zpráva	9	3.2

gen_1	2403	2.0
ublížení	168	10.52
krádež	164	9.82
výtržnictví	64	9.61
zneužívání	94	9.42
poškozování	66	9.38
podvod	113	9.34
porušování	88	9.33
vydírání	65	9.26
zpronevěra	52	9.15
maření	47	9.14
vlastizrada	46	9.08
zneužití	72	9.08
hanobení	44	9.01
loupež	55	8.98
týrání	37	8.53
vražda	107	8.52
pomluva	34	8.51
šíření	57	8.44
padělání	27	8.35
kuplířství	24	8.29
ohrožení	52	7.95
krácení	22	7.89
podplácení	17	7.78
znásilnění	20	7.68
ohrožování	16	7.66

Fig. 1. A part of the WS Table of *čin (act)* in SYN2000

3.2 Terminological Verbs

Together with already mentioned gen_1 table we also obtained Sketch tables containing several verbs with legal meaning. Some of them are listed in Table 3. In [1] a group of verbs collected from legal texts was investigated. They represented a mixture of the various common verbs and also some legal ones. In comparison with them the verbs obtained now from the Sketch tables are actually terminological verbs with legal meaning. Some of them could be straightforwardly used as entries in a dictionary of legal terms.

Legal verbs were explored in [2] and they were added to the lexical database VerbaLex [9]. In this way the Verbalex was extended with a reasonable number of the legal verbs. However, it is still possible to find candidates of legal verbs for further extension of VerbaLex in our Word Sketch tables (e.g. *promlčet(be time-barred)*, *překvalifikovat (change qualification)*, *prošetřovat (investigate)*, *zpochybňovat (question)*,...).

Table 3. Legal verbs from word sketch tables

spáchat (commit)	dopouštět se (perpetrate)
páchat (commit)	odsoudit (condemn, sentence)
vyšetřovat (investigate)	potrestat (punish)
překvalifikovat (change qualification)	přiznat (confess)
prošetřovat (investigate)	postihovat (affect)
vykonat (perform)	prokázat (prove)
zmařit (thwart)	ohlásit (announce)
ospravedlňovat (justify)	uprchnout (escape)
stíhat (prosecute)	zodpovídat (be responsible)
objasňovat (explain)	zadržet (arrest, detain)
promlčet (be time-barred)	napravit (amend)
zpochybňovat (question)	litovat (regret)

In Sketch tables we, of course, find also other parts of speech but they usually do not contain any data relevant for legal terminology (prepositions, particles, etc.). However, there is one more interesting Sketch table that should be mentioned—the one with adjectives. While adjectives are not typical dictionary entries, some of them should be explained at least in a hierarchical context of the headword *čin (act)* (e.g. *úmyslný (deliberate)*, *nedbalostní (caused by negligence)*, *násilný (violent)*, *protiprávní (illegal))*.

3.3 Verbs with Financial Meaning

Verbs explored in [2] also include a group of verbs occuring in legal text and belonging to the financial domain. While the verbs mentioned above were processed by the WSE here we decided to have a look at the verbs in whose complex valency frames the argument labeled as EXT(sum:1) occurs. Then we explored their frequencies in the corpus Czes (see Table 4 on the next page).

First, it has to be remarked that the verbs in the list fall into small subgroups containing semantically close items – they are either aspect pairs or even triples, if iteratives are considered. We will not deal with the pairs perfective : imperfective here, the category of aspect belongs to the area of morphology in Czech.

It can be observed that the differences in the frequencies of the particular verbs in the table are significant. It is not difficult to conclude that the less frequent verbs in the list display specialized terminological meanings, for

Table 4. Examples of Financial Verbs

Verb	frequency in Czes
alokovat (allocate)	670
realokovat (reallocate)	13
danit (tax)	45,374
zdanit (tax)	3,291
dodanit (pay up the tax)	117
dodaňovat (pay up the tax)	20
dlužit (owe, have a debt)	11,773
vydlužit (take on loan)	13
fakturovat (invoice)	755
vyfakturovat (invoice)	135
financovat (finance)	18,598
dofinancovat (finance up)	219
předfinancovat (prefinance)	19
počítat (calculate, compute)	116,673
spočítat (calculate, compute)	13,663
tarifikovat (tariff)	23
tarifovat (tariff)	12
validovat (validate)	65
valorizovat (valorise)	591
platit (pay)	290,253
zaplatit (pay up)	148,683
splatit (pay off)	10,787
proplatit (pay out, cash)	2,070
proclít (clear through customs)	119
vyclít (clear through customs)	8
vydražit (auction off)	2,465
vydražovat (be auctioning)	11

instance *vyclít (clear through customs)* with the frequency 8 or *předfinancovat (prefinance)* with 19. We are aware that the frequency cannot serve as the only convincing indicator of the terminological status of these verbs – more detailed evaluation would be needed. In any case, for the verbs in the Table 4 we can say that the ones with the frequency lower than 1,000 can be reliably considered terminological.

4 Verbalex Valency Frames and Legal Terms

In this section we will briefly touch the relation between complex valency frames of the financial verbs as they can be found in VerbaLex and their corresponding Word Sketches obtained from the corpus Czes. The assumption is that the semantic labels of the verb's arguments such as AG(person:1|institution:1) or EXT(sum:1) match reasonably with the concrete nouns that appear in the Word Sketch tables of the respective verbs.

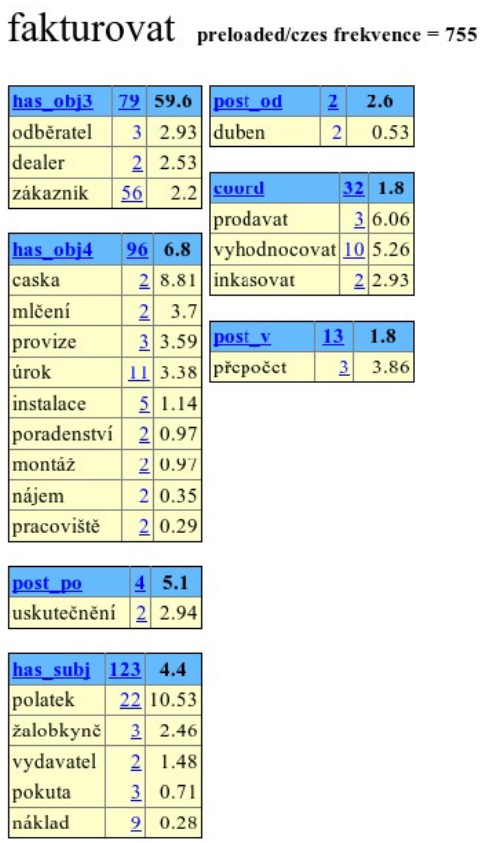


Fig. 2. WS of *fakturovat (invoice)* in Czes

Take, for instance, the complex valency frame for the verb synset *vy/fakturovat (invoice)* capturing its financial meaning:

1: fakturovat$_{n1}$, zaúčtovat$_{n1}$, zaúčtovávat$_{n1}$, naúčtovat$_{n1}$, naúčtovávat$_{n1}$
AG<person:1>$^{obl}_{kdo1}$ VERB^{obl} REC<person:1|institut.:1>$^{opt}_{komu3}$ ART<goods:1>|ACT<act:2>$^{opt}_{za+co4}$ EXT<sum:1>$^{obl}_{co4}$
–example: advokátka si fakturovala za své služby desetitisíce korun (impf)
–example: obsluha hostovi zaúčtovala stopadesátikorunový poplatek (pf)
–example: číšník zákazníkovi naúčtuje špatnou cenu (pf)
–synonym:
–use: prim
–reflexivity: obj_dat

In the valency frame of the verb *fakturovat (invoice)* we find the arguments labeled as AG<person:1>, REC<person:1|institut.:1>, ART<goods:1>, ACT<act:2>, EXT<sum:1>. The question thus is whether they have real counterparts (tokens) in the Word Sketch (Table 2). The simple manual comparison

shows that the answer is positive and that nouns found in the respective corpus sentences semantically agree with what is predicted by the argument labels in the valency frame. We think that it is not not necessary to go into details here but the next step should consist in an attempt to formulate a formal procedure that would perform exactly this.

It has to be remarked that in corpus sentences we observe that some of the arguments are frequently expressed by personal pronouns (mostly classified as subjects and objects), thus we should be able recognize that e.g. personal pronouns *já, ty, on, mu (I, you, he, him)* match with labels like AG(person:). These cases have to be handled by a procedure defined just for this purpose. The next phenomenon that we must deal with are passive verbs forms, which, as we can see in the corpus, because of their transitivity transform the order of the arguments and their surface valencies, i.e. they (in Czech) substitute the accusative case with nominative or instrumental with nominative. The verbs in VerbaLex contain information about their transitivity or intransitivity but we need to formulate transformation rules which will do this automatically when it is needed – the passive verb forms then will serve as a trigger.

Another phenomenon that plays a relevant role in this context are anaphora relations and their resolution. The frequency of the personal pronouns that function as antecedents in anaphoras is quite high, for instance the frequency of *já (I)* in the corpus Czes is 1,981,248, the frequency of *on (he)* is 5,726,584, thus role of the anaphorical relations cannot be neglected. Unfortunately, the present versions of the algorithms handling the resolution of the anaphorical relations in Czech are not successful enough for the indicated task. It also has to be taken into account that processing of the personal pronouns by the Sketch Engine is still at its beginning. The handling of the demonstrative pronouns is also relevant in this respect and we are afraid that it is even more difficult task.

5 Conclusions

In this paper we have discussed some possible techniques for semiautomatic finding the terminological entries that could be used in building the Czech legal term dictionary. We have investigated the behaviour of the noun candidates of legal terms using the Word Sketch Engine and can conclude that the obtained results are promising though, at the moment, we cannot offer a complete quantitative evaluation. We also explored some verbs with financial meaning – for them we found that their frequencies in the corpus Czes convincingly prove their terminological nature. At the end we have briefly dealt with the relation between complex valency frames of the financial verbs as they can be found in VerbaLex and their corresponding Word Sketches obtained from the corpus Czes. This comparison shows that in this way it is possible to obtain more detailed information about the meaning of the verbs belonging to the financial domain but not only for them. These observations can be generalized also for the verbs from other domains.

Acknowledgements

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536 and in the National Research Programme II project 2C06009 and by the Czech Science Foundation under the projects P401/10/0792 and 407/07/0679.

References

1. Pala, K., Rychlý, P., Šmerk, P.: Automatic Identification of Legal Terms in Czech Law Texts. In: Semantic Processing of Legal Texts, Berlin, Springer (2010) 83–94.
2. Pala, K., Mráková, E.: Verb Valency Frames in Czech Legal Texts. In: Proceedings of the RASLAN 2009 Workshop, Brno, Masaryk University (2009).
3. Cvrček, F.: Právní informatika (Legal Informatics). Publisher A. Čeněk, Plzeň (2010).
4. Mráková, E.: Partial Syntactic Analysis (of Czech). Ph.D. thesis, Faculty of Informatics, Masaryk University, Brno (2002) (in Czech).
5. Kilgarriff, A., Rychlý, P., Smrž, P., Tugwell, D.: The Sketch Engine. In: Proceedings of the Eleventh EURALEX International Congress, Lorient, France: Universite de Bretagne-Sud (2004) 105–116.
6. Pala, K., Rychlý, P.: A Case Study in Word Sketches – Czech Verb *vidět (see)*. In: A Way with Words: Recent Advances in Lexical Theory and Analysis (A Festschrift for Patrick Hanks), Menha Publishers (2010) 187–198.
7. Ústav Českého národního korpusu: Korpus SYN2000. `http://ucnk.ff.cuni.cz/syn2000.php` (2000).
8. Natural Language Processing Centre, FI MU: Czes Corpus. `http://corpora.fi.muni.cz/ske/auth/` (2009).
9. Horák, A., Hlaváčková, D.: VerbaLex – New Comprehensive Lexicon of Verb Valencies for Czech. In: Computer Treatment of Slavic and East European Languages, Third International Seminar, Bratislava, VEDA (2005) 107–115.

CzechParl: Corpus of Stenographic Protocols from Czech Parliament

Miloš Jakubíček and Vojtěch Kovář

Natural Language Processing Centre, Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
xjakub@fi.muni.cz, xkovar3@fi.muni.cz

Abstract. Within a single language, there is a large variety of styles used for written text and speech, which differ significantly and have their subtle specifics. Among all of those, the language of politicians represents an integral class that deserves detailed analysis. In this paper we present CzechParl, a corpus we built from stenographic protocols recorded during plenary meetings of the Czech parliament in its modern era from 1993 to 2010. We provide brief statistics of the corpus and discuss its intended future usage and further development.

1 Introduction

During the last century language became the main weapon of politicians in modern democratic countries, directed to the citizens (voters) which are exposed to it in everyday rush. Often it is political speech that changes their opinions and heavily influents elections results. Obviously this leads to situations when not only pure informative and communicative, but also demonstrative, manipulative and psychological language functions are exhibited, sometimes in a latent form that comes mostly unnoticed. It is therefore of great importance that also this kind of language becomes subject to linguistic analysis and introspection.

For this purpose one needs data that will be large enough to be representative. One straightforward source of political texts represent various news and newspapers that focus on public life. Unfortunately most of those are not available in electronic archives with free access, moreover all of them do not contain texts crafted by politicians directly, but rather comments and glosses written by journalists and also lots of non-political texts – and as such they are not suitable for the projected needs.

Therefore we turned our attention to the place where everyday politics is being performed, namely the Czech parliament, and where we can benefit from the fact that for legal reasons stenographic protocols of politicans' speeches are made during all plenary meetings and are freely available in the electronic form.

In further text we describe CzechParl, a corpus built from such stenographic protocols (known as Hansards in United Kingdom and other Commonwealth countries) recorded in both chambers of Czech parliament – the Chamber of Deputies (1993–2010) and Senate (1996–2010). We briefly refer on how the corpus

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 41–46, 2010.

was prepared, report on its structure and basic statistical characteristics and discuss further analysis that is going to be performed in the future.

2 Corpus Building

2.1 Joint Czech and Slovak Digital Parliamentary Library

In 1993 the Joint Czech and Slovak Digital Parliamentary Library [1] (further referred to as JCSDPL) has been announced, a shared initiative of Czech and Slovak parliaments which aimed at providing free access to both modern and historical parliamentary documents in electronic form. It contains documents since 1848 that were produced in several legislative institutions [2], besides these, the Czech (Bohemian) Assemblies Digital Library [3] was built on top of the JCSDPL, providing historical documents dated back to the 11th century. The institutions covered by JCSDPL are as follows:

- Austrian Constituent Imperial Diet 1848–1849 (Vienna, Kromeriz)
- Diet of the Czech Kingdom 1861–1913
- National Assembly of the Czechoslovak Republic and the Czechoslovak Socialist Republic 1918–1968
- Diet of the Slovak Republic 1939–1945
- Slovak National Council 1944–1960
- Czech National Council 1969–1992
- Resolutions of the presidium of the Slovak National Council 1970–1987
- Federal Assembly of the Czechoslovak Socialist Republic and the Czechoslovak Federal Republic (Chamber of the People and Chamber of Nations) 1969–1992
- Parliament of the Czech Republic (Chamber of Deputies and Senate) since 1993
- National Council of the Slovak Republic since 1993

Following types of documents are part of the JSCDPL:

- Invitations for sessions
- Debates
- Bills
- Resolutions
- Materials of committees

In the current work we have processed the protocols from the modern era of the Czech parliament (since 1993) that contain documents in Czech only and are of most interest with regard to the intended analyses: the debates that are stenographically recorded and as such represent a unique source of truly captured discourses.

While the JSCDPL definitely represents an invaluable language resource, it was not intended for any automatic processing or annotation. Historical documents are available in the form of scanned images, modern ones (including those very recent that have been processed) as HTML pages. Therefore extensive cleaning and post-processing was needed to obtain plain text accompanied with the desired annotation (as described in further text).

```
 <p><s><speech name="Miroslav Kalousek" role="Poslanec"> : Nebojte se
, já nechci reagovat na pana poslance Ratha . </s><s> Jsou příspěvky ,
na které se dá reagovat pouze nonverbálně a to mi Schwarzenberg zakázal
. </s><s> ( Ohlas . ) </s></p><p><s> Rád bych ale zareagoval na pana
poslance Sobotku a odmítl jeho tvrzení , že z větší části mé vystoupení
nesouviselo s projednávaným návrhem . </s><s> Pokud jste nepochopil
přímou souvislost mnou prezentovaných indikátorů s vaším návrhem , pak
se nedivím , že ten návrh předkládáte , pane poslanče . </s><s> Příště
prosím nechte své svědomí plout po vlnách mých vět a otevřete srdce
mým slovům a poznáte pravdu . </s><s> ( Pobavení v pravé části sálu .
) </s></speech></p>
```

Fig. 1. Random sample of corpus annotation.

2.2 Annotation Scheme

The source data have been converted from HTML into plain text, relieved of any boilerplate (e. g. HTML-related metadata like headers and footers) and afterwards tokenized, segmented into sentences and lemmatized as well as tagged using the `desamb` tagger [4] which works on top of the morphological analyser of Czech called `majka` [5,6].

Furthermore, following structures have been annotated using an XML-like markup (as given in parentheses):

- **sentences** (`<s>`)
- **paragraphs** (`<p>`), as given in the stenographic protocols
- **discourses** (`<speech>`), extracted from the stenographic protocols and containing the speaker name and role
- **meeting days** (`<day>`), containing the date of the meeting
- **documents** (`<doc>`), where each document represents an electoral term of either the Chamber of Deputies or Senate

A random sample of corpus source text is provided in Figure 1. Next, the corpus data in such form have been encoded using the Manatee/Bonito corpus management system [7,8], components empowering the Sketch Engine [9], enabling fast and effective search and analysis including lookup of individual speeches by speaker name or advanced querying using the Corpus Query Language [10,11].

3 CzechParl Statistics

The corpus is available for view and search upon registration on `http://corpora.fi.muni.cz`. Bonito, the web interface built on top of Manatee, enables submitting of powerful queries and creating sophisticated statistical reports. A screenshot of the web interface is provided in Figure 2.

In Table 1, a summary of basic statistical properties of CzechParl is provided. Notable is the significant difference in the size between the part originating in

Table 1. Statistical summary of attributes and structures present in CzechParl.

Parliament chamber	Chamber of Deputies	Senate	Total
Tokens	75,050,917	6,823,205	**81,874,122**
Sentences	3,987,910	198,816	**4,186,726**
Paragraphs	1,549,717	70,655	**1,620,372**
Documents	9	7	**16**
Days	1,985	140	**2,125**
Discourses	85,983	5,964	**91,947**

the Chamber of Deputies and the part recorded in Senate: over 90% of the corpus comes from the Chamber of Deputies. This corresponds to the hypothesis that most political debates occur in Chamber of Deputies, which also convenes more often than the Senate.

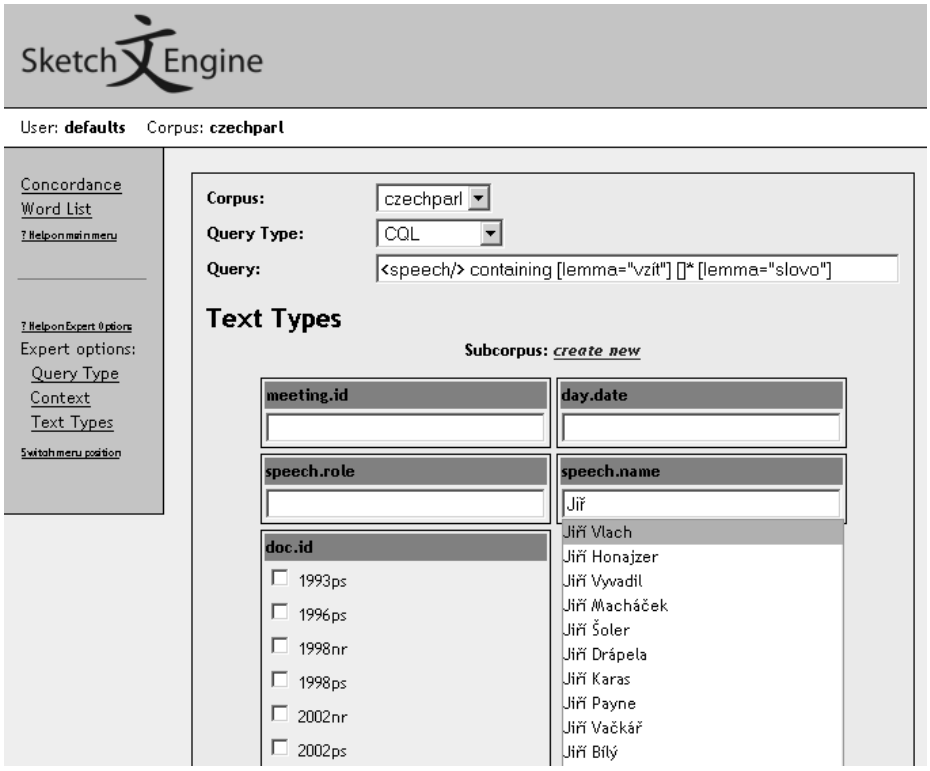


Fig. 2. Screenshot of the Bonito query interface.

4 Related Work

Similar attempts to build corpus of parliamentary documents have been performed for Dutch [12] and Spanish [13]. An online demo for searching small part (1994–1997) of German parliamentary documents is available as well as part of the Corpus Workbench project [14]. An important resource in this domain is also the EuroParl [15], a parallel corpus of documents originating in the European parliament, which however focuses on a different goal, namely statistical machine translation.

Even though parliamentary documents in most countries are publicly available (including stenographic protocols), the prevailing majority still waits for being processed into the form of an annotated and searchable text corpus that will encourage researchers to provide corpus-based evidence for their theories on political discourse, following the notable example of [16] where corpus-motivated studies in this domain are presented for 11 European parliaments.

5 Conclusions and Future Development

In this paper we presented CzechParl, a corpus of parliamentary documents from both chambers of the modern Czech parliament – the Chamber of Deputies and Senate. Source texts have been obtained from the Joint Czech and Slovak Digital Parliamentary Library. The corpus contains annotation of individuals speeches and as such it is suitable for further linguistic analysis and introspection focused on political discourse.

In particular we plan in the future the corpus to be subject to analysis with regard to what Just calls "floscula", a sort of thought-terminating clichés that lost their original meaning and suite to fool their readers/hearers: "Floscula is – be it deliberate, intentional and malae fidei (propaganda, ideology, advertisement, kitch) or subconscious, automatic and mechanic (style, trend, snobbish slang) – hiding and decorating of emptiness by words". [1]A dictionary of flosculae compiled by Just [17] represents an excellent basis for such analyses.

Since the collected corpus data contains over 100 millions of tokens, the content of CzechParl is also going to become a part of `czes` [18], a big Czech web corpus that is currently under development.

Acknowledgements

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536 and in the National Research Programme II project 2C06009 and by the Czech Science Foundation under the project P401/10/0792.

[1] In Czech original: „Floskule je – ať už záměrné, účelové a obmyslné (propaganda, ideologie, reklama, kýč), nebo podvědomé, automatické a mechanické (móda, trendovost, snobský slang) – zakrývání, zdobení prázdna slovy." [17]

References

1. Parliamentary Library, Information Technology Department of the Office of the Chamber of Deputies, Information Technology Department of the Senate Chancellery of the Czech Parliament, Parliamentary Library, parliamentary archive and Information Technology Department of the National Council of the Slovak Republic: Joint Czech and Slovak Digital Parliamentary Library. [online] `http://www.psp.cz/cgi-bin/eng/sqw/hp.sqw?k=82` (2003) [cit. 16. 11. 2010].
2. Parliamentary Library, Information Technology Department of the Office of the Chamber of Deputies, Information Technology Department of the Senate Chancellery of the Czech Parliament, Parliamentary Library, parliamentary archive and Information Technology Department of the National Council of the Slovak Republic: Joint Czech and Slovak Digital Parliamentary Library. [online] `http://www.psp.cz/cgi-bin/eng/eknih/info.htm` (2003) [cit. 16. 11. 2010].
3. Parliamentary Library of the Czech Republic: Czech (Bohemian) Assemblies Digital Library. [online] `http://www.psp.cz/cgi-bin/eng/sqw/hp.sqw?k=82` (2003) [cit. 16. 11. 2010].
4. Šmerk, P.: Unsupervised Learning of Rules for Morphological Disambiguation. In: Lecture Notes in Computer Science, Springer Berlin / Heidelberg (2004).
5. Šmerk, P.: Towards Computational Morphological Analysis of Czech. Ph.D. thesis, Faculty of Informatics, Masaryk University, Brno (2010).
6. Šmerk, P.: Fast Morphological Analysis of Czech. In: Proceedings of the RASLAN Workshop 2009, Brno (2009).
7. Rychlý, P.: Korpusové manažery a jejich efektivní implementace. Ph.D. thesis, Fakulta informatiky, Masarykova univerzita, Brno (2000).
8. Rychlý, P.: Manatee/Bonito – A Modular Corpus Manager. In: 1st Workshop on Recent Advances in Slavonic Natural Language Processing, Brno (2007) 65.
9. Kilgarriff, A., Rychlý, P., Smrž, P., Tugwell, D.: The Sketch Engine. In: Proceedings of EURALEX. (2004) 105–116.
10. Christ, O., Schulze, B.M.: The IMS Corpus Workbench: Corpus Query Processor (CQP) User's Manual. University of Stuttgart, Germany, `http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench` (1994).
11. Jakubíček, M., Rychlý, P., Kilgarriff, A., McCarthy, D.: Fast syntactic searching in very large corpora for many languages. In: PACLIC 24 Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation, Tokyo (2010) 741–747.
12. Marx, M., Schuth, A.: DutchParl. A Corpus of Parliamentary Documents in Dutch. In: Proceedings of LREC 2010. (2010) `http://politicalmashup.nl/dutchparl`.
13. Martin, C., Marx, M.: Parliamentary documents from Spain. In: Proceedings of LREC 2010. (2010) `http://politicalmashup.nl/SpanishParliament`.
14. CWB open-source community: `BUNDESTAG` corpus. [online] `http://cogsci.uni-osnabrueck.de/~korpora/ws/CQPdemo/Bundestag/` (2010) [cit. 16. 11. 2010].
15. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: MT Summit. (2005).
16. Ilie, C., ed.: European Parliaments Under Scrutiny: Discourse Strategies and Interaction Practices. John Benjamins Publishing, Amsterdam, Netherlands (2010).
17. Just, V.: Velký slovník floskulí. LEDA, Praha (2009).
18. Horák, A., Rychlý, P., Kilgarriff, A.: Czech word sketch relations with full syntax parser. In: After Half a Century of Slavonic Natural Language Processing, Brno, Czech Republic, Masaryk University (2009) 101–112.

Utilizing Linguistic Resources
Theory and Practical Experience

Vašek Němčík

NLP Laboratory
Faculty of Informatics, Masaryk University
Brno, Czech Republic
xnemcik@fi.muni.cz

Abstract. The Prague Dependency Treebank (henceforth PDT) is a large collection of texts in Czech. It contains several layers of rich annotation, ranging from morphology to deep syntax. It is unique in its size and theoretical background, especially for a language like Czech, which can be, with regard to the number of its speakers, considered a small language. In this article, we use PDT 2.0 to demonstrate that within real NLP systems, complex annotations may cut both ways. We present several issues that might pose problems when extracting data from PDT, and complex structures in general, and hint on possible solutions.

1 Introduction

The Prague Dependency Treebank 2.0 (henceforth just PDT) is a large collection of Czech texts compiled at the Institute of Formal and Applied Linguistics at the Charles University in Prague. It is probably the most notable linguistic resource available for the Czech language. Taking into account that Czech is a rather "small" language by the number of its speakers, PDT can be considered unique as it exhibits a very flattering combination of large corpus size and annotation richness. The main aim of the work on PDT was to yield a resource that would allow testing the theoretical claims following from the long tradition of the Functional Generative Description of language by confronting them with real data. Further motivation was to obtain data for training machine-learning based NLP applications.

Without any doubt, the emergence of PDT has made it possible to study linguistic phenomena that were not easy to investigate on a large scale before. On the other hand, with all respect to the long-standing tradition of FGD and the work done on PDT, it can be argued that the potential of PDT as training corpus for real NLP systems could be further extended by simplifying the data structures.

In the next section, we overview The Prague Dependency Treebank and its main features. Next, in Section 3, we mention selected features of the PDT annotation that might be considered rather unfortunate for practical purposes, as illustrated on a particular NLP task. Finally, we conclude the paper by reviewing the presented ideas and sketching the plans of further work.

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 47–51, 2010.

2 The Prague Dependency Treebank

The PDT is an open-ended project for manual annotation of substantial amount of Czech-language data with linguistically rich information ranging from morphology through syntax and semantics/pragmatics and beyond [1]. Its central point is a large corpus containing the mentioned annotation, now available in version 2.0. It is probably the most notable linguistic resource available for Czech. Taking into account its size and richness of annotation, and a rather limited number of speakers of Czech, it can be even cosidered unique.

The information about texts in PDT is organized as multi-layer annotation. The overview of layers available is presented in Table 1.

Table 1. The layers in PDT 2.0

annotation layer	*brief characterization*	*size*
morphological layer	(words and morphological features)	2 million words
analytical layer	(syntactic dependency trees)	1.5 million words
tectogrammatical layer	(trees with deep sentence structure)	0.8 million words

The first layer of annotation, **the morphological layer**, contains information about grammatical features of individual words. For each sentence in the corpus, it contains a linear list of words, accompanied by their respective morphological tags. More information about the features and their possible values within this layer can be found in [2].

The analytical layer comprises of the same tokens as the previous layer, however, their organization is not linear. On this layer, they form a tree based on the relation of syntactic dependency.

The nodes of the trees (i.e. the individual words) contain information about further features. The most relevant is probably the analytical function describing the grammatical role of the word (subtree) in the sentence. Each node also carries information about its linear position in the sentence, and also a link to the related token on the morphological layer. The linking between the a-layer and m-layer is one-to-one, i.e. each node on the a-layer has a corresponding token on the m-layer, and vice versa. More information about the analytical trees can be found in [3].

The highest level level of PDT 2.0, **the tectogrammatical layer**, captures diverse linguistic aspects beyond syntax. Each sentence is represented by a dependency tree reflecting the deep structure of each sentence.

The nodes of the tectogrammatical tree carry various further information. Each node carries its semantic role with regard to its structural mother, and where applicable, nodes carry information about its valency. Further, tectogrammatic trees contain information about grammatemes of auto-semantic words, topic-focus articulation, coreference, etc. Notably, the linking between

the t-layer and a-layer is *not* one-to-one. Many nodes of the analytical layer have been ommitted (e.g. prepositions and punctuation), on the other hand, new nodes have been added (e.g. representations of people or things that are semantically present, but not explicitly voiced in the sentence). More information on features stored in tectogrammatical trees is revealed in [4].

The annotation was carried out manually, based on outputs of various automatic tools that yield an approximate form of the relevant information.

3 Practical Issues

As mentioned above, PDT is a very enticing linguistic resource, both in its size and the scope of phenomena it encompasses. On the top of that, it is supported by the underlying Functional Generative Description theory, which has a long and respectable tradition in general linguistics, and there is hardly any doubt about its consistency.

On the other hand, designing successful NLP systems often requires a rather different, practical, and sometimes even slightly heretic, approach to language. It is not crucial that the system is based on a sophisticated linguistic theory, and that it handles marginal phenomena correctly, as long as it performs reasonably with regard to its purpose. This concerns both the algorithm design, and the data used within the system. Usually it is of great advantage when the underlying principles are rather straightforward.

This is obviously the case with the notion of syntactic dependency which is central in the Praguian linguistic tradition. The notion of one word being syntactically dependent on some other word, is computationally very feasible, and also the theoretical consequences are very straightforward.

The dependency trees within the PDT, however, are not plain dependency trees. Apart from dependency edges, they also contain edges of various other types. These edges mainly account for coordination and apposition. At first glance, this seems to be a very clean and elegant solution, however, together with the convention of attaching arguments of coordinated nodes to the respective conjuction, it alters the tree structure considerably. Most importantly, it has a rather unfortunate consequence, namely that unlike in a plain dependency tree, a phrase is not necessarily a (sub)tree. This fact makes processing of the tree data rather cumbersome.

This can be demonstrated on a sample (yet very real) processing task – detection of unvoiced subjects of clause predicates. PDT seems to be a suitable source of training or evaluation data for this task. However, extracting this type of data from PDT is not as straightforward as it may seem.

Firstly, PDT does not explicitly contain information about clauses. This seems to be a consequence of the fact that the notion of clauses is somewhat irrelevant from the dependency point of view. Unfortunately, for many NLP tasks, such as re-construction of missing subjects in pro-drop languages, it is the main processing unit.

The next step and a logical way towards our goal would be to detect clauses and their predicates based on the information stored on the analytical layer. This can be done procedurally, by traversing a-layer trees in a top-down manner. However, this process is rather cumbersome. Verbal nodes representing a clause predicate are not easy to distinguish from infinitives as the relevant auxiliary verbs, modal verbs, and other relevant nodes might possibly be at various positions of the tree. So might be the node representing the subject, the presence (or absence) of which is the key point of our investigation. As a result of this, we arrive at a heuristic procedure detecting clause boundaries and missing subjects, with a non-zero error rate.

This is rather disappointing, as the information we need to extract from PDT seems to be a key factor for various decisions during the annotation process, both on the analytical, and the tectogrammatical layer. In practice, a comparably feasible alternative to extracting this information from PDT would probably be computing this information from plain text using shallow parsing and simple heuristics.

The use of information on the tectogrammatical layer in real-life NLP systems lies probably in the future as most of the data can't be obtained automatically with a satisfactory reliability by contemporary systems. However, a considerable obstacle in practical usability of t-layer trees seems to be their rather complex structure. Apart from the constructions common on the analytical layer, there are further phenomena that have an impact on the basic notion of dependency in the t-layer trees, such as several types of newly generated trees and linkings. Studying the representational conventions of the tectogrammatical layer to prevent unexpected results, is a rather time-consuming task as the available annotation manual consists of 1215 pages. Unfortunately, this fact as such means a significant motivation to search for alternative data sources. It also inevitably raises the question whether it is possible for a human, as error-prone as they are in their essence, to produce consistent and reliable annotation based on such large and complex annotation guidelines. These psychological effects are rather unfortunate as these doubts are probably hollow.

This is an interesting contrast to projects such as The Sketch Engine, which is based on simple, however, from the linguistic point of view not particularly clean ideas. The contrast suggests that also in the world of language technology, simplicity is at least as appealing as a wide range of features.

4 Conclusions and Further Work

This paper reviewed the main features of The Prague Dependency Treebank, and its annotation levels. Further it described certain difficulties that may arise when using complex linguistic data in a practical NLP setting.

The presented obstacles in extracting a specific type of information from PDT hints that richness of data structures is not always a clear advantage. A stricter (simpler) implementation of the dependency principle within the tree structures might make data easier to use. As our future work, we plan to refine

our heuristics for extracting clauses and unvoiced clause subjects from the PDT annotation and to export it into a simple, linear token-based format.

Acknowledgements

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536.

References

1. Hajič, J., et al.: The Prague Dependency Treebank 2.0. Developed at the Institute of Formal and Applied Linguistics, Charles University in Prague. (2005) `http://ufal.mff.cuni.cz/pdt2.0/`.
2. Zeman, D., Hana, J., Hanová, H., Hajič, J., Hladká, B., Jeřábek, E.: A Manual for Morphological Annotation, 2nd edition. Technical Report 27, ÚFAL MFF UK, Prague, Czech Republic (2005).
3. Hajič, J., Panevová, J., Buráňová, E., Urešová, Z., Bémová, A.: Anotace Pražského závislostního korpusu na analytické rovině: pokyny pro anotátory. Technical Report 28, ÚFAL MFF UK, Prague, Czech Republic (1999).
4. Mikulová, M., Bémová, A., Hajič, J., Hajičová, E., Havelka, J., Kolářová-Řezníčková, V., Kučová, L., Lopatková, M., Pajas, P., Panevová, J., Razímová, M., Sgall, P., Štěpánek, J., Urešová, Z., Veselá, K., Žabokrtský, Z.: Anotace Pražského závislostního korpusu na tektogramatické rovině: pokyny pro anotátory. Technical report, ÚFAL MFF UK, Prague, Czech Republic (2005).

Frequency of Low-Frequency Words in Text Corpora

Pavel Rychlý

Natural Language Processing Centre, Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
pary@fi.muni.cz

Abstract. Low-frequency words, esp. words occurring only once in a text corpus, are very popular in text analysis. Also many lexicographers draw attention to such words. This paper lists a detailed statistical analysis of low-frequency words. The results provides important information for many practical applications, including lexicography and language modeling.

1 Introduction

Text Corpora play a crucial role in current linguistics. They provide empirical data for studies on how a language is used. Almost any analysis of a text corpus uses some form of statistics and raw frequency of words is the most common.

Correct handling of the most frequent words is very important in many applications. Other applications ignore most frequent words, usually listed in a stop-list, and pay more attention to mid-frequent or low-frequent words.

There are linguistics studies where the key role of a text analysis have words occurring only once in the whole text (corpus). Such words are called *hapax legomena*. The related terms *dis legomenon*, *tris legomenon*, and *tetrakis legomenon* refer respectively to double, triple, or quadruple occurrences, but are far less commonly used.

This paper quantifies how important very low frequent words could be. It also discusses frequency distribution of zero-frequency words.

2 Corpora

To test frequency distributions of different phenomena we have build a set of corpora, each corpus with different size. Corpora was created by repeated random selection of a text unit from a master Corpus. The Manatee system [1] was used for the random selection of corpus parts and creation of subcorpora. The results presented in this paper use the British National Corpus (BNC) [2] as the master corpus.

We have tested the following units, each corresponds to a XML tag in the BNC source data: sentences (`<s>`), paragraphs (`<p>`), texts (`<text>` – this one covers only written part of the corpus, one document could be divided into several texts), and documents (`<bncdoc>`).

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 53–57, 2010.

Choosing smaller units for sampling results in more random corpus. On the other hand, bigger units creates corpora containing more natural texts. The selection of unit size make a big difference in frequency distributions of low-frequency words. Figures 1 and 2 shows percentages of hapax legomena in

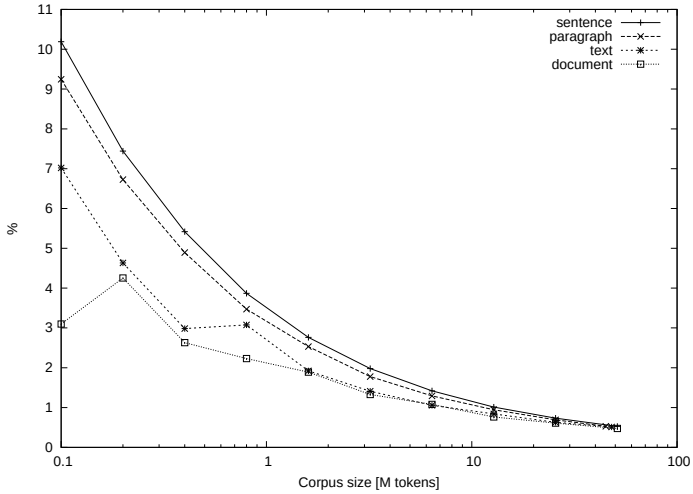


Fig. 1. Percentage of text covered by hapax legomena. Comparison of four sampling units

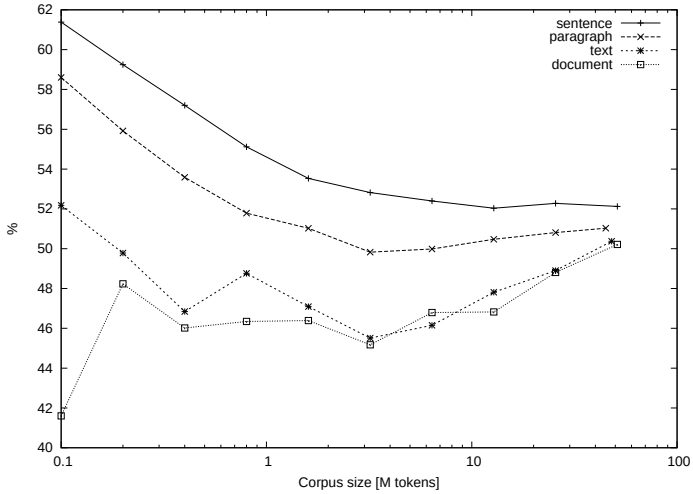


Fig. 2. Percentage of word types (size of the corpus lexicon) of hapax legomena. Comparison of four sampling units

corpora of different size and using different sampling units. First is based on token numbers, second on word types.

The size of a sampling unit is not important for high-frequency words, Figure 3 displays percentage of text which is covered by words occurring exactly 10 times in the corpus. We can see that even small corpora (200,000 tokens) have no difference in coverage for different sampling units.

3 Low-Frequency Words in Corpora

The important question about low-frequency words is how frequent are hapax legomena from a corpus measured in the whole language. We will simulate "the whole language" by much bigger corpus. For this purpose we have chosen sequence of 3 samples of the BNC, starting with 1 million tokens and following with 10 million and 100 million tokens.

The Figures 7, 6 and 8 show frequency distribution of words with selected fix frequency (hapax or tris legomena) in ten times bigger corpus. In all the graphs x-axis lists all different frequencies in which any selected word occurs in the bigger corpus. The y-axis then measures how many of selected words have such feature. For example, a point at $[x = 10, y = 15]$ means, there are 15 different words which occurs exactly 10 times in the bigger corpus. Words are selected from the smaller corpus and occurs there exactly ones or three times (depending on the graph).

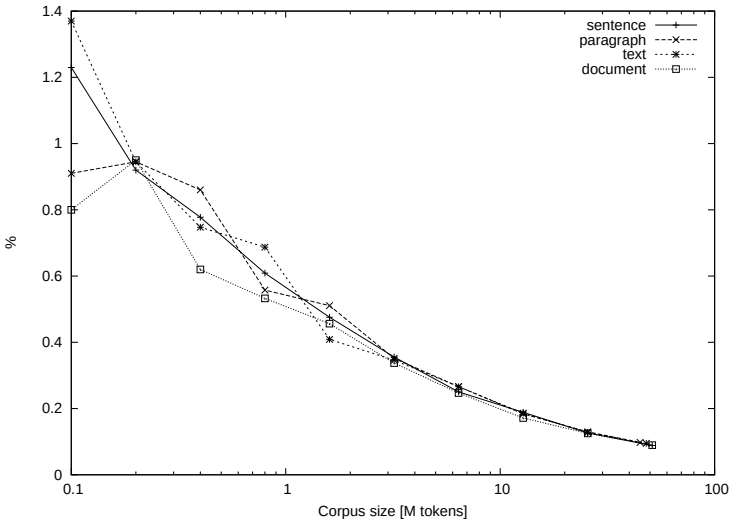


Fig. 3. Percentage of text covered by words occurring exactly 10 times in the corpus. Comparison of four sampling units

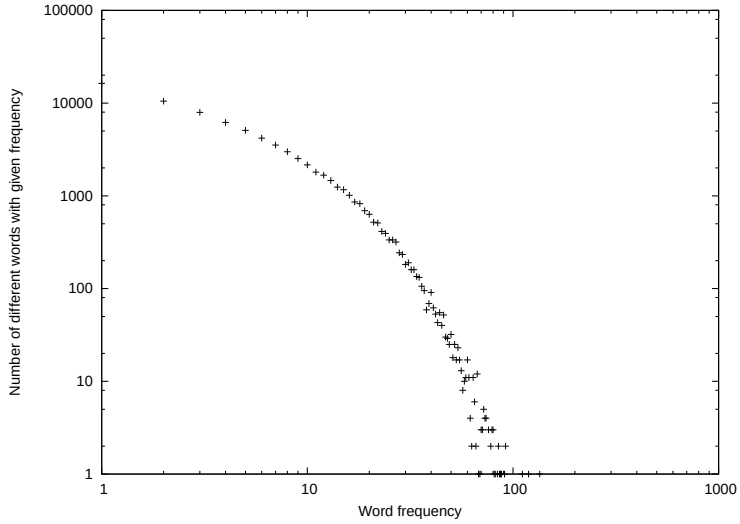


Fig. 4. Frequency of hapax legomena form 10M corpus in 100M corpus. Sentence as an sampling unit

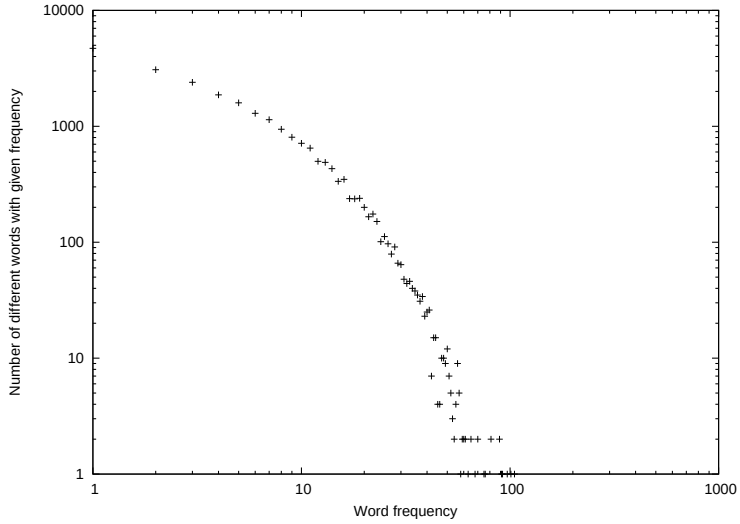


Fig. 5. Frequency of hapax legomena form 1M corpus in 10M corpus. Sentence as an sampling unit

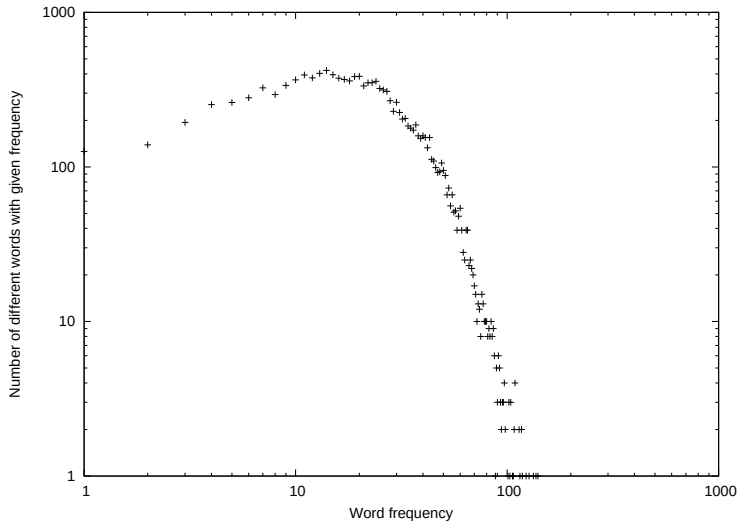


Fig. 6. Frequency of tris legomena form 10M corpus in 100M corpus. Sentence as an sampling unit

4 Conclusion

Even though Low-frequency words form a big part of a corpus lexicon (hapax legomena about 50%), they are not significant in the text. Especially big corpora provide enough data for all relevant words.

Sampling unit have big impact on the frequency distribution of low frequency words. Selection of only sentences or paragraphs instead of whole texts or documents can destroy frequency distribution of words, esp. low-frequency ones.

Acknowledgements

This work has been partly supported by the Ministry of Education, Youth and Sports of Czech Republic under the project LC536 and within the National Research Programme II project 2C06009, and by the Czech Grant Agency under the projects P401/10/0792 and 407/07/0679.

References

1. Rychlý, P.: Manatee/Bonito–A Modular Corpus Manager. RASLAN 2007 Recent Advances in Slavonic Natural Language Processing (2007).
2. Aston, G., Burnard, L.: The BNC handbook: exploring the British National Corpus with SARA. Edinburgh Univ Pr (1998).

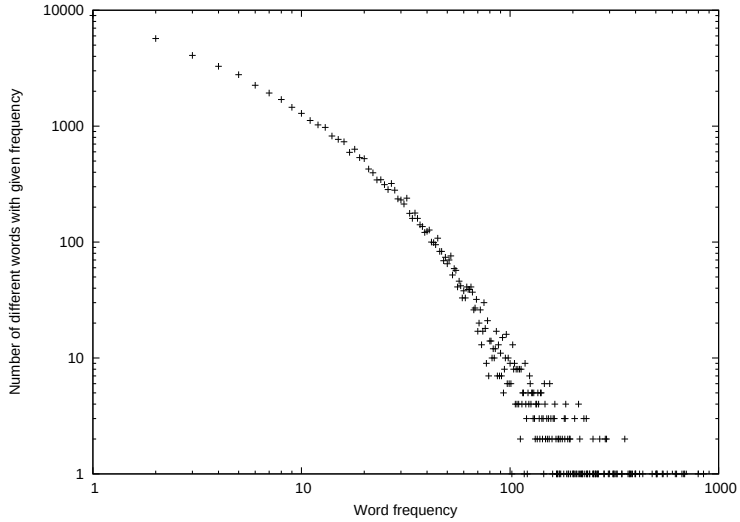


Fig. 7. Frequency of tris legomena form 10M corpus in 100M corpus. Document as an sampling unit

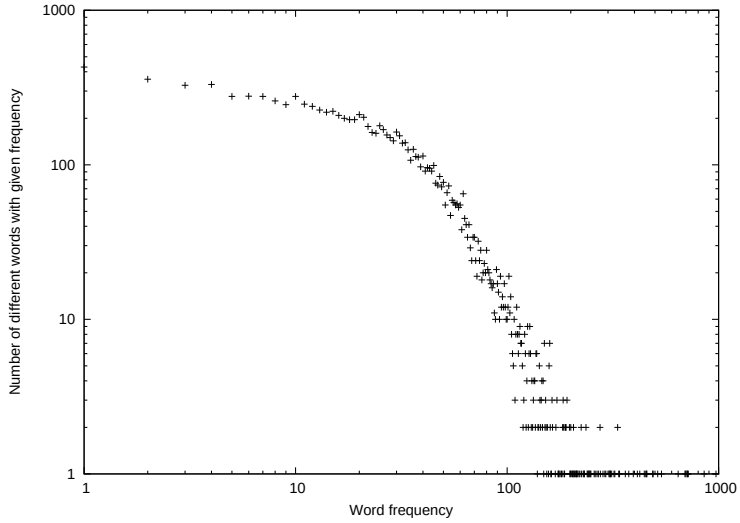


Fig. 8. Frequency of hapax legomena form 10M corpus in 100M corpus. Document as an sampling unit

Part III

Logic in Language

Time Aspects of Transparent Intensional Logic in Communication and Decision-Making of Agents

Jakub Macek and Tomáš Frydrych

VŠB-Technical University Ostrava
17. listopadu 15, 708 33 Ostrava, Czech Republic
jakub.macek.st@vsb.cz frydrych.t@gmail.com

Abstract. This paper shows pitfalls in the formal representation of a state of a multiagent system that is changing in time and we propose a solution based on Transparent intensional logic. Moreover, we propose a method for the definition of facts and rules and communication between agents in such a way that preserves consistency of agents' knowledge base. Using an example we present a set of methods for information retrieval and nondeterministic decision-making proces. These methods can be used by an agent that is entering the system with limited knowledge in order to reach a specified goal.

Key words: TIL, multiagent system, agent, communication

1 Introduction

Transparent Intensional Logic (TIL) [1] is a logical system founded by Prof. Pavel Tichý [4,5]. It is a higher-order system primarily designed for the logical analysis of natural language. As an expressive semantic tool it has a great potential to be utilised in artificial intelligence and in general whenever and wherever people need to communicate with computers in a human-like way.

Due to its rich and transparent procedural semantics, all the semantically salient features are explicitly present in the language of TIL constructions. It includes explicit intensionalisation and temporalisation, as well as hyper-intensional level of algorithmically structured procedures (known as TIL constructions) which are of particular importance in the logic of attitudes, epistemic logic and the area of knowledge representation and acquisition. We use TIL constructions for agents communication and decision-making proces.

Imagine an agent agent in a multiagent system [2,3]. The agent is able to make independent decisions and these decisions together with the rules defined in the system can cause the system transition from one state to another. The goal of the agent is to achieve a predefined state of the system.

In this paper we use a demo example of a game in order to illustrate our method and the proposed solutions. The system is a game of dungeon exploration and contains one or more independent players who know the rules of the game at the very beginning. The game consists of several places connected

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 61–67, 2010.

with pathways (thus resembling undirected graph). Cumulative information about position of all players in game is its state.

Players is able to make only one kind of a decision, namely where to move next. As a consequence of these decisions the game gradually changes its state as the players are moving around.

The key part of the game that moves it closer to a real life situation is that players have limited knowledge of the system. Each decision helps them obtain new information along the way with some of their knowledge becoming obsolete over time.

Beside agent-players who are capable of decisions we have another type of agent. These represent local sources of information and player agents simulate perceiving their surroundings by communicating with them.

2 Solution without Time

To show insufficiency of the solution without using temporal aspect of the problem, let's define more simpler game with limited scope. We will use just one agent-player (denoted by Z) and a map consisting of two places (A and B). The player will be able to move between these two places without restriction. We will introduce appropriate rules for the player movements into the system. We also have to include a rule stating that the player can be at one place at a time.

At the beginning, the player will be at place A. Knowledge base of game thus contains these facts and rules:

$$\forall x((LocatedAt(x,A) \wedge MovesTo(x,B)) \supset LocatedAt(x,B))$$
$$\forall x((LocatedAt(x,B) \wedge MovesTo(x,A)) \supset LocatedAt(x,A))$$
$$\forall x(LocatedAt(x,A) \supset \neg LocatedAt(x,B))$$
$$\forall x(LocatedAt(x,B) \supset \neg LocatedAt(x,A))$$
$$LocatedAt(Z,A)$$

And as a result of the last fact, we can also deduce that:

$$\neg LocatedAt(Z,B)$$

Now the player makes a decision to move in the place B:

$$MovesTo(Z,B)$$

The system infers which is the new position of the player:

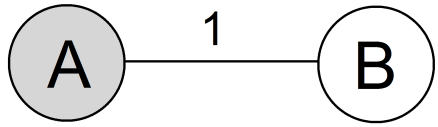
$$\forall x((LocatedAt(x,A) \wedge MovesTo(x,B)) \supset LocatedAt(x,B)) \wedge MovesTo(Z,B)$$
$$\models LocatedAt(Z,B)$$

We can now see that this conclusion contradicts the assumption that $\neg LocatedAt(Z,B)$. Obviously, the problem is due to the fact that we did not take into account temporal aspects of agents positions. This can be solved by removing invalid facts from knowledge base, but a more elegant solution is to introduce time to our formal representation of the system.

3 Introduction of Temporal Aspect to the Knowledge Base

Seemingly contradictory states can be avoided by introducing temporal aspect. Transparent intensional logic (TIL) uses real numbers for time variables, but for illustration we will limit ourselves in this paper to integers (representing individual seconds). For example the moment $t_2 = t_1 + 1$ is one second after the moment t_1. Additionally we define constant ${}^0Beginning \to \tau$, which will represent the first moment of the life cycle of our multiagent system.

First we define a define basic system with one agent Z and two places A, B that are linked through pathway 1 and are 30 seconds apart. Its obvious that the agent starting at the place A will move towards the place B and reach it at Beginning + 30.



The initial state of the system is constructed by the following construction:

$$\lambda w[[[{}^0LocatedAt\ w]\,{}^0Beginning]\,{}^0Z\,{}^0A]$$

Arguments of the function $LocatedAt/((o\iota)\tau)\omega)$ are an agent and a place, respectively.

Moreover, we introduce the rule for a movement of any agent $x \to \iota$ at time $t \to \tau$, which will represent moving through the only available pathway.

$$\lambda w\lambda t\lambda x[{}^0\supset\ [[[{}^0LocatedAt\ w]\ t]\ x\,{}^0A][[[{}^0LocatedAt\ w]\ [{}^0+\ t\,{}^030]\ x\,{}^0B]]$$

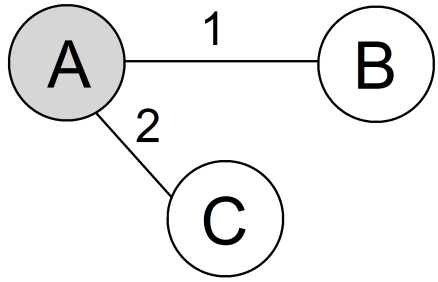
In natural language we can read this rule as follows: If at time t an agent x is at the place A then at time $t + 30$ the agent x will be at the place B.

Using this rule and information about its initial position, the agent can now deduce that after 30 seconds it will be located at place B.

$$\lambda w[[[{}^0LocatedAt\ w]\ [{}^0+\ {}^0Beginning\,{}^030]]\,{}^0Z\,{}^0B]$$

4 Decisions

In the previous section we considered a trivial situation in which the agent has no option and must move from the place A to the place B. Now let the rule that an agent cannot stay in the same place be valid and we add another possible destination C such that C is connected with A and the transition from A to C takes 20 seconds.



It is clear that we introduced some form of nondeterminism into the system. We need to compensate this by the concept of decision. This will be formalised by the following construction:

$$\lambda w[[[^0 Decision\ w]\ ^0 Beginning]^0 Z\ ^0 1]$$

Type $Decision/(((0\iota)\tau)\omega)$ where the first argument is an agent and the second argument is a pathway.

Now the rule describing a movement from one place to another can be expanded by adding a condition specifying that the agent decided to go through pathway 1. In the same way we create another rule for pathway 2.

$$\lambda w\lambda t\lambda x[^0\supset\ [^0\wedge[[[^0 LocatedAt\ w]\ t]\ x\ ^0 A][[[^0 Decision\ w]\ t]^0 Z\ ^0 1]]$$

$$[[[^0 LocatedAt\ w]\ [^0+\ t\ ^0 30]\ x\ ^0 B]]$$

$$\lambda w\lambda t\lambda x[^0\supset\ [^0\wedge[[[^0 LocatedAt\ w]\ t]\ x\ ^0 A][[[^0 Decision\ w]\ t]^0 Z\ ^0 2]]$$

$$[[[^0 LocatedAt\ w]\ [^0+\ t\ ^0 20]\ x\ ^0 C]]$$

We need to properly inform the agent about the need to make a decision and the possible options. For the place A that means to offer the agent to use either pathway 1 or 2. In order to comply with physical laws, we also specify that the agent can make only one of these decisions at a particular time t.

$$\lambda w[^0\supset\ [[[^0 LocatedAt\ w]^0 Beginning]\ ^0 Z\ ^0 A]$$

$$[^0\vee[[[^0 Decision\ w]\ ^0 Beginning]^0 Z\ ^0 1][[[^0 Decision\ w]\ ^0 Beginning]^0 Z\ ^0 2]]]$$

$$\lambda w\lambda t\lambda x[^0\supset\ [[[^0 Decision\ w]\ t]x\ ^0 1][^0\neg[[[^0 Decision\ w]\ t]x\ ^0 2]]$$

$$\lambda w\lambda t\lambda x[^0\supset\ [[[^0 Decision\ w]\ t]x\ ^0 2][^0\neg[[[^0 Decision\ w]\ t]x\ ^0 1]]$$

The decision made by the agent is a new fact in the knowledge base. Nevertheless we must ensure that agents do not pollute the system knowledge base with contradictory facts by specifying which facts can be introduced. For instance, we can introduce a constraint that only facts generated from a particular scheme not contradicting agent's knowledge are decisions. In our example

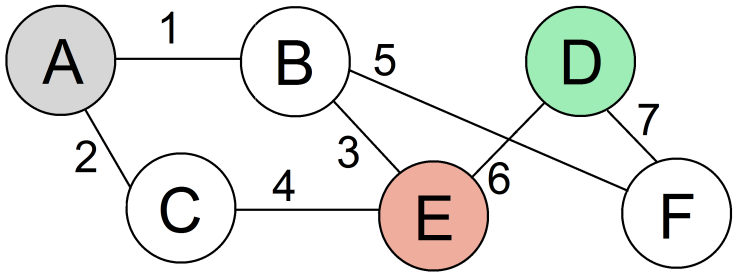
the scheme can be: "any proposition containing only function *Decision* with parameters: current world as w, current time as t, me (agent-player) as x and any pathway individual as remaining argument".

For example if an agent is located at the place B at time t, then it cannot go through the pathway 2, since it is not available. In the same way we could introduce temporary unavailability of the pathway 1 from the place A (with regard to time t at which the decision is being made).

5 Simulation

So far we have defined agent-player that is capable of random decisions based on its knowledge base and blind exploration of the game system. Of course, such behavior is not very useful.

Let us extend the definition of our system so that the system behaviour is closer to the dungeon exploration. We will add more places and define positive and negative goals to motivate the agent. Positive goal (finding the tresure) will be defined as "agent is at place D". Negative goal (meeting a grue or falling into a chasm) will be defined as "agent is at place E".



$$\lambda w \lambda t \lambda t_1 \lambda x [{}^0\supset\ [{}^0\wedge [[[{}^0 LocatedAt\ w]\ t]\ x\ {}^0B][{}^0\geq\ t_1\ t]]$$

$$[[[{}^0 PositiveGoal\ w]\ t_1]\ x]]$$

$$\lambda w \lambda t \lambda t_1 \lambda x [{}^0\supset\ [{}^0\wedge [[[{}^0 LocatedAt\ w]\ t]\ x\ {}^0C][{}^0\geq\ t_1\ t]]$$

$$[[[{}^0 NegativeGoal\ w]\ t_1]\ x]]$$

These propositions representing goals have to affect the decision-making process. Decisions leading to positive goals will be favored, whereas decisions leading to negative goals will be avoided. However we must take into account that there is a possible situation in which every decision will lead to a negative goal. Therefore we cannot just dismiss such decisions, because the agent would run out of possible options and that would contradict previously defined rule.

In order that the agent knows the effects of its decisions these decisions must be simulated. To preserve consistency of information in knowledge base each simulation will have a separate simulated knowledge base. At the beginning of any simulation the simulated knowledge base will contain everything that is in real knowledge base (ie. information about agent's position, rules of movement,

etc.), but all facts deduced during simulation will be stored only in the separate knowledge base.

The depth of simulation (meaning maximum number of simulated decisions in one simulation) will have direct influence on agent's intelligence. However the agent can make only one decision at time, because the state of the system in future is not guaranteed to be the same as in a simulation (and most likely will not be). To determine which possible decision is the best one, the agent needs to properly rank each one according to outcomes of simulations.

Decisions that ended with a positive goal in the first step (after the first decision) are considered to be optimal. If a positive goal has been reached later in the simulation, the decision should be ranked as slightly less optimal. Immediate decisions which didn't lead to a positive or negative goal are considered to be neutral, and finally decisions leading to a negative goal are ranked as the least optimal.

Immediate decision that ranked best will be done by the agent (meaning introduced both to its own and global knowledge base) and proper consequences will be deduced. In our example the consequence would be agent's position in following moments.

6 Agent-Informer and Communication

At the beginning the agent-player doesn't know all facts and rules describing the system. In fact, such a situation is not actually possible, because agent would be by definition omniscient. Instead the agent knows only those rules that apply now and here. It can obtain this information by communication with immobile agents-informers.

In reality this agent-informer would be agent-player's ability to get additional information by observing its surroundings. In our example the primary information is the list of possible decisions, in particular possible destinations. Agent-player will keep this information in its knowledge base.

To perform sufficiently extensive simulation, the agent needs also information about other places (in other words it needs to create a temporary map of nearby places). This information can be provided by agent-informer, too, but only for actual moment. It cannot provide reliable information about future state of the system. If agent obtained at place A information about pathway 5 that always leads from place B and when at place B obtained information that pathway 5 is temporarily unavailable, it would create a contradition in its knowledge base.

Information about future is always only assumed and solution to mentioned problem is to treat it as such. When agent-informer gives assumptions to agent-player, it will limit them to simulated environment and possibly also limit their usablity to a specific time span.

Simulation will be represented with special fact $Simulation/(o\iota)$, which will be true only if agent given as parameter is performing a simulation. For example, agent Z will introduce following fact to its simulated knowledge base:

$[{}^0Simulation\ {}^0Z]$. Using this fact as a condition we can transfer assumptions from agent-informer to agent-player the same way as real facts.

7 Conclusion

In this paper we demonstrated a method for the definition of facts, rules and communication between agents in the multiagent system that is changing in time and presents solution using Transparent intensional logic. Using an example we presented a set of methods for information retrieval and nondeterministic decision-making proces.

Acknowledgements

This research has been supported by the Grant Agency of the Czech Republic, project 401/09/H007 'Logical Foundations of Semantics', and also by the internal grant agency of VSB-TU of Ostrava – SP/2010214 Modeling, simulation and verification of software processes II.

References

1. Duží, M., Jespersen, B., Materna, P.: *Procedural Semantics for Hyperintensional Logic (Foundations and Applications of TIL)*. Springer. 2010. 17. Logic, Epistemology, and the Unity of Science.
2. Duží, M., Ciprich, N., Košinár, M., Kohut, O., Frydrych, T.: *The Architecture of an Intelligent Agent in MAS*. Waki Print Pia, Kanagawa, Japan. 2008.
3. Kohut, O., Košinár, M., Frydrych, T.: *TIL in Knowledge-Based Multi-Agent Systems*. Masarykova universita. 2008. 11.
4. Tichý, P. *The Foundations of Frege's Logic*. Walter de Gruyter, Berlin-New York, 1988.
5. Tichý, P. (2004): *Collected Papers in Logic and Philosophy*, V. Svoboda, B. Jespersen, C. Cheyne (eds.), Prague: Filosofia, Czech Academy of Sciences, and Dunedin: University of Otago Press.

How to Analyze Natural Language with Transparent Intensional Logic?

Vojtěch Kovář, Aleš Horák, and Miloš Jakubíček

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{xkovar3, hales, xjakub}@fi.muni.cz

Abstract. Logical analysis of natural language text is generally an underspecified task. However, within the project aiming at automatic processing of natural language (NL) text by means of logical analysis followed with the inference process, we need to have a "de facto" standard way for analysis of each NL sentence. First steps for introducing such standard are described in the presented text.
The paper describes a semi-automatic way of building a corpus of logic formulae (constructions) in the formalism of the Transparent intensional logic (TIL) for real-world sentences in the Czech language. Output of a syntactic parser is used to determine the logical structure of the sentence and a verb valency lexicon is exploited for assigning TIL types. Using this information, an exemplary bank of TIL constructions is created automatically. This corpus of TIL constructions is then checked by human logic experts who iteratively consult the results with the respective theory of TIL transcription and the processing of input supportive lexicons. A user-friendly interface for such checking is presented at the end of the paper.

1 Introduction

Formalization of natural language utterance is one of the most important steps in automatic natural language understanding. Successful specification of this process is a necessary assumption to intelligent handling of natural language texts, ranging from text classification or intelligent information extraction to question answering systems. Since formal logical systems are one of the bases of theoretical computer science, they are well known and described and the representation of natural language sentences in a logical formalism may help automatic programs to work with "meaning" in natural language.

The respective logical system we are dealing with in this paper is called the Transparent Intensional Logic (TIL [1]), an expressive logical system introduced by Pavel Tichý [2], which works with a complex hierarchy of types, system of possible worlds and times and an inductive system for derivation of new facts from a knowledge base in development.

In the current project, we aim at development of a syntactic analyzer for Czech with the ability of generating TIL logical constructions from the parsing

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 69–76, 2010.

results (according to Horák's *Normal Translation Algorithm* [3]). During the process of standardization of natural language analysis in TIL, an exemplary corpus of TIL constructions, as a result of logical analysis of real-world sentences, is being created in a semi-automatic way that can be later used for human reference as well as training and evaluating future knowledge representation and reasoning tools.

2 Related Work

Published results in the area of natural language (NL) logical analysis regarding algorithms for converting natural language sentences into logical formulae cover mostly the First Order Predicate Logic [4,5]. However, it can be shown that first-order formalisms are not able to handle systematically the NL phenomena like intensionality, belief attitudes, grammatical tenses and modalities (modal verbs and modal particles in natural language). On the other hand, since TIL works with techniques designed for capturing the natural language meaning these problems either do not arise or they can be solved in an intuitive way in TIL (see [2]).

For the Czech language, no attempt of logical analysis of real-world NL texts is known at the time. There is a language abstraction known as tectogrammatical layer [6] (which is the result of work at the Institute of Formal and Applied Linguistics in Prague) and there are attempts to obtain this tectogrammatical layer automatically [7]. However, the obtained results are so far not complete enough and also the tectogrammatical layer of language description cannot be really called a logical formalism.

3 The TIL Formalism

The theory of the Transparent Intensional Logic (TIL) is formed by a higher-order logical system, which uses an extended type hierarchy. TIL was first introduced by its creator Pavel Tichý in [2] and then specified in numerous articles and books. From the last publications, we refer especially to [8] and [1].

In TIL, the meaning of a natural language expression is described as a *construction* procedure, which describes in a creative way the "definition" of the expression subject. The notation of these procedures uses a λ-calculus formalism – "*red apple*" is analysed as a class of individuals $\lambda w \lambda t \lambda x.[\mathbf{red}_{wt}\, x \wedge \mathbf{apple}_{wt}\, x]$.

The TIL type hierarchy is built over a type base of four types:

- o (omicron) – truth values *True* and *False*
- ι (iota) – class of (labels of) individuals
- τ (tau) – class of real numbers/time moments
- ω (omega) – class of possible worlds

These types can be combined to mappings in order to form all types of order 1. Together with classes of constructions organized by the order of their sub-constructions, they allow us to refer to objects of higher-order (type of order n denoted as $*_n$).

Variables are the only simple constructions, complex constructions are created inductively by the following simple operations:

- *trivialization* – construction constructs the trivialized object
- *abstraction/closure* – construction of a (classic) function
- *application/composition* – construction of functional application
- *execution* and *double execution* – construction of the result of functional application(s)

Constructions that contain only objects of types of order 1 are denoted as *constructions of order 1*. Constructions of order n together with types of order n form a class of objects of order $n + 1$.

These intuitive rules of the extended type hierarchy allow us to refer to all complicated phenomena in natural language, such as belief attitudes or procedural definitions.

4 The Synt Parser

The parser used in the mentioned project is called `synt` [9]. It is based on a large CFG grammar with contextual actions (ca. 3,500 rules generated from 200 meta-rules) and an efficient variant of the head-driven chart parsing algorithm. As its input, `synt` takes a morphologically (ambiguously) annotated Czech sentence. The possibly ambiguous parsing result can be represented in various formats: output trees, a much more compact packed forest of these trees, the analysis chart, list of extracted noun and prepositional phrases, clauses, a dependency graph and some others. Moreover, the output parsing trees are ranked according to their probability and there is an algorithm for efficient obtaining one or more best unambiguous analyses.

The resulting parsing trees can be converted into the formulae in the TIL formalism, using the algorithm described in [3] including the prototype implementation. The algorithm basically „consists in assigning the appropriate (sub)constructions to analysed (sub)constituents by employing the lexicon and in the type checking which makes it possible to prune the contingencies that cannot be resolved on a lower level of the derivation tree" [3].

Therefore, for wide coverage of natural language texts we need to build a lexicon of TIL types, namely for verbs and their arguments since they contain the essential information of the sentence. This is further described in the next section.

5 Building the Verb Lexicon

As explained in [3], we need to have a high-coverage lexicon of TIL types of Czech verbs. For building such a lexicon, we have used the valency lexicon of the Czech verbs *VerbaLex*.

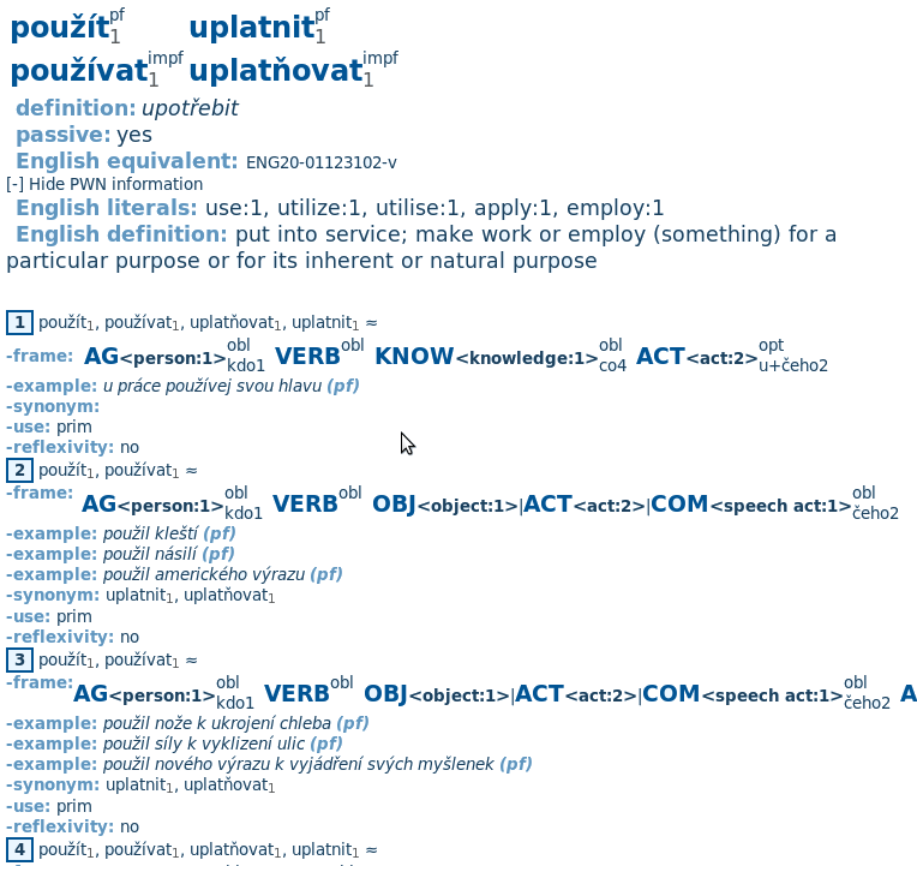


Fig. 1. Example of VerbaLex complex valency frames for 'používat'

5.1 VerbaLex

VerbaLex [10] is an exhaustive lexicon of Czech verb valency frames, currently containing over 10,000 Czech verbs (see Figure 1). The valency information is given in the form of syntactic (shallow) valencies, i.e. the morphological categories that a phrase needs to meet to be able to be an argument of a particular verb, as well as in the form of semantic valencies, i.e. semantic classes that a particular phrase usually belongs to in order to form a valency of that verb.

The semantic classes used in *VerbaLex* are compatible with the ones used in the Czech *WordNet* [11] so that it is relatively easy to find a semantic class of a given word and check if it fits to the valency frame of a particular verb. This is of course limited by the (Czech) *WordNet* coverage, precision of the data in both *WordNet* and *VerbaLex* and the fact that long phrases that cannot be found in *WordNet* can also stand as verb arguments in the sentence.

The information in the lexicon may also be used for pruning ambiguous syntactic analysis (by omitting analyses producing different verb phrases than the one recorded in the lexicon). This is already implemented and used in the `synt` system for the shallow valencies [12].

5.2 From VerbaLex to the Lexicon of the TIL Types

To build the lexicon of the verb types as outlined in Section 4, we have exploited the information from the *VerbaLex* lexicon. The number of arguments can be obtained from the obligatory members of the verb frame and their types can be derived from their semantic roles which is present in the valency lexicon as well.

Translating semantic roles of the possible verb arguments into the TIL types clearly needs a mapping from the semantic roles to the types. The most recent work is aimed at building such a mapping – an introductory analysis of this task was published in [13].

The resulting lexicon of the verb types is then given as the parameter to the `synt` parser that builds the corresponding TIL construction from the most probable syntactic tree for the sentence, according to the Normal Translation Algorithm. If the analysis produces a valid construction, this construction is then included into the corpus to be evaluated by human logical experts, as described in the next sections.

6 Creating the Corpus

The corpus of TIL constructions is built on top of the morphologically annotated corpus DESAM [14]. The unambiguous morphological annotation (manually checked by linguistic experts) will minimize the errors on the morphological analysis level and therefore the overall result quality will be better (compared to using e.g. ambiguous or automatically disambiguated morphological information).

Furthermore, in the initial stage of the project we confine ourselves to analyze simple sentences in present, past and future tense, which means that the whole sentence contains exactly one clause with exactly one verb. Such sentences are supposed to be handled well by the parser and the selection will therefore help to further reduce number of errors in the constructions generated automatically.

In the future, the logical analysis in `synt` will cover more complicated phenomena, such as relative subordinate sentences or complex sentences with temporal events including direct speech.

7 Evaluating the Automatic Constructions

As mentioned above, all constructions on the output of the parser are included into the corpus of constructions. However, such a resource contains a lot of errors that may come from various levels of the analysis. Therefore, logic specialists are asked to provide feedback to the parser developers so that errors can be fixed in the right places.

Fig. 2. Corpus of TIL constructions – detailed view of a selected sentence

7.1 The TIL Corpus Web Interface

For this purpose, a web interface for the corpus was developed[1] to make the work of the logic specialists easier and to provide a visualisation of correct, incorrect and not yet checked constructions.

As can be seen in Figure 2, each sentence is displayed as the TIL construction (automatically created by the parser) as well as the plain text that enables easy reading. On the right side of each sentence, there are two buttons for marking the constructions accepted or wrong. There is also a status indicator showing if the sentence has already been checked and what was the decision.

If the decision was negative, the user (logic specialist) is asked to provide a brief description of the error. The decisions are stored in the database as well as the error descriptions and information about the user; all of this is immediately available to the parser developers. Also, the history of changes is kept by the means of the `git` versioning system.

[1] `http://corpora.fi.muni.cz/til`

8 Conclusions

In the paper, we have described the first steps to create an exemplary corpus of natural language data annotated for their logical structure. As a process parallel to the corpus creation, the parser producing the logical formulae is being improved using the feedback from corpus human experts. At the end of this process, we hope to have a high-quality corpus of TIL constructions as well as a wide-coverage parser producing TIL constructions with a reasonable precision.

The work has however just been started. The future will hopefully bring intensive development of the parser including adapting it to more complicated sentences as well as incremental increase of the quality of the corpus.

Acknowledgements

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536 and in the National Research Programme II project 2C06009 and by the Czech Science Foundation under the project P401/10/0792.

References

1. Duzı, M., Jespersen, B., Materna, P.: Procedural Semantics for Hyperintensional Logic. Foundations and Applications of Transparent Intensional Logic. Volume 17 of Logic, Epistemology and the Unity of Science. Springer, Berlin (2010).
2. Tichý, P.: The Foundations of Frege's Logic. de Gruyter, Berlin, New York (1988).
3. Horák, A.: The Normal Translation Algorithm in Transparent Intensional Logic for Czech. Ph.D. thesis, Faculty of Informatics, Masaryk University, Brno (2002).
4. Pease, A., Li, J.: Controlled english to logic translation. Theory and Applications of Ontology: Computer Applications (2010) 245–258.
5. Yusuke Miyao, Alastair Butler, K.Y., Tsujii, J.: A Modular Architecture for the Wide-Coverage Translation of Natural Language Texts into Predicate Logic Formulas. In: Proceedings of Pacific Asia Conference on Language, Information and Computation, PACLIC 2010, Institute for Digital Enhancement of Cognitive Development, Waseda University (2010) 481–488.
6. Mikulová, M. e.a.: Annotation on the Tectogrammatical Level in the Prague Dependency Treebank: Annotation Manual. Universitas Carolina Pragensis (2008).
7. Klimeš, V.: Transformation-based tectogrammatical analysis of Czech. In: Text, Speech and Dialogue, Springer (2006) 135–142.
8. Tichý, P.: Collected Papers in Logic and Philosophy. Prague: Filosofia, Czech Academy of Sciences, and Dunedin: University of Otago Press (2004).
9. Horák, A., Kadlec, V.: New Meta-grammar Constructs in Czech Language Parser synt. In: Lecture Notes in Artificial Intelligence, Proceedings of Text, Speech and Dialogue 2005, Karlovy Vary, Czech Republic, Springer-Verlag (2005) 85–92.
10. Hlaváčková, D., Horák, A.: VerbaLex – New Comprehensive Lexicon of Verb Valencies for Czech. In: Proceedings of the Computer Treatment of Slavic and East European Languages 2005, Bratislava, Slovakia (2005) 107–115.
11. Pala, K., Smrž, P.: Building the Czech WordNet. Romanian Journal of Information Science and Technology **7**(2–3) (2004) 79–88.

12. Jakubíček, M.: Enhancing Czech Parsing with Complex Valency Frames. Master's thesis, Masaryk University, Brno (2010).
13. Horák, A., Pala, K., Duží, M., Materna, P.: Verb Valency Semantic Representation for Deep Linguistic Processing. In: Proceedings of the Workshop on Deep Linguistic Processing, ACL 2007, Prague, Czech Republic, the Association for Computational Linguistics (2007) 97–104.
14. Pala, K., Rychlý, P., Smrž, P.: DESAM — annotated corpus for Czech. In: Proceedings of SOFSEM '97, Springer-Verlag (1997) 523–530 Lecture Notes in Computer Science 1338.

Process Ontology

Marie Duží[1], Martina Číhalová[1], Marek Menšík[1,2], and Lukáš Vích[1]

[1] Department of Computer Science FEI, VSB-Technical University Ostrava
17. listopadu 15, 708 33 Ostrava, Czech Republic
[2] Institute of Computer Science FPF, Silesian University Opava
Bezručovo nám. 13, 746 01, Opava, Czech Republic
m.tina.cihal@gmail.com, marie.duzi@vsb.cz, mensikm@gmail.com,
lukas.vich.st@vsb.cz

Abstract. In the paper we propose a method of building up ontology of processes. First, a summary of our approach to ontology building using Transparent Intensional Logic (TIL) is presented. Since TIL operates smoothly at three levels of abstraction, namely hyperintensional, intensional and extensional level, we have a logical machinery to explicate concepts as hyperintensions which we define as TIL closed constructions in their normal form. Constructions are *procedures* that produce intensions or extensions as their products. Thus TIL is apt for modelling ontology as an extensional *logic of (hyper-)intensions*. However, while ontologies of a given specific domain are frequently studied, ontology of *processes* has been rather neglected. The goal of this paper is to fill the gap and to extend our method to the specification of process ontology. Since *logical* analysis presupposes full linguistic competency, we intend to make use of the results of linguistic analysis, in particular of verb-valency frames. Each process can be specified by a verb (what is to be done), possibly with parameters like the agent/actor of the process (who), the object to be operated on, resources, etc. Since valency frames correspond to senses of verbs, we thus obtain a finer specification of the process/procedure. Particular properties of the actor and other parameters are then specified as requisites of the process or its typical properties.

Key words: til, ontology, valency frames, process

1 Introduction

The term 'ontology' has been borrowed from philosophy, where ontology is a systematic account of existence. In recent computer science and artificial intelligence a formal ontology is an explicit and systematic conceptualization of a domain of interest. Given a domain, ontological analysis should clarify the *structure* of knowledge on what exists in the domain. A formal ontology is, or should be, a stable heart of an information system that makes knowledge sharing, reuse and reasoning possible. As J. Sowa says in [9], "logic itself has no vocabulary for describing the things that exist. Ontology fills that gap: it is the

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 77–88, 2010.

study of existence, of all the kinds of entities — abstract and concrete — that make up the world".

Current languages and tools applicable in the area of an ontology design focus in particular on the *form* of ontological representation rather than what a *semantic content* of ontology should be. Of course, a unified syntax is useful, but the problems of syntax are almost trivial compared to the problems of developing a common semantics for any domain. Duží and Materna in [3] dealt with semantic ontology content in general. In this paper we focus on a process ontology design. Each process can be specified by a verb (*what* is to be done), with parameters like the agent/actor of the process (*who*), the object to be operated on, resources, etc. Since valency frames correspond to senses of verbs, we thus obtain a specification of the process. As a specification tool we apply the *procedural semantic* framework of Transparent Intensional Logic (TIL) which is briefly introduced in Section 2. Section 3 provides a summary of an ontology content in general, and in the main Section 4 we propose a process ontology based on valency frames. The proposal is illustrated by an example of the analysis of an accident.

2 TIL in Brief

Since the pioneering paper [5] logicians and semanticists have striven to define so-called *structured meanings* that would comply with the principles of compositionality and universal referential transparency. Various adjustments of Frege's semantic schema have been proposed, shifting the entity named by an expression from the extensional level of atomic (physical/abstract) objects to the intensional level of molecular objects such as sets or functions/mappings. Yet natural language is rich enough to generate expressions that talk neither about extensional nor intensional objects. Propositional attitudes are notoriously known as the hard cases that are neither extensional nor intensional, as Carnap in (1947) [1] characterized them. It has become increasingly clear since the 1970s that we need to individuate meanings more finely than by possible-world intensions, and the need for hyperintensional semantics is now broadly recognised. Our position is a plea for *hyperintensional* semantics, which takes expressions as encoding *algorithmically structured procedures* producing extensional/intensional entities (or lower-order procedures) as their products. This approach — which could be characterized as being informed by an algorithmic or computational turn — has been advocated by, for instance, Moschovakis in (1994) [8]. Yet much earlier, in the early 1970s, Tichý introduced his notion of *construction* and developed the system of Transparent Intensional Logic (TIL).[3]

Constructions, as well as the entities they construct, all receive a type. The ontology of TIL is organized in an infinite, bi-dimensional hierarchy of types. Since we strictly distinguish between a construction of an object and the object itself, and between a function and its value, construction must be always of a higher order than the object it constructs, and a function is of a higher degree

[3] See Tichý (1988 and 2004) [11,12].

than its value. Thus one dimension of the type hierarchy increases molecular complexity of functions, the other dimension increases the order of constructions. Our definitions are inductive, and they proceed in three stages. First, we define the simple types of order 1 comprising non-constructions. Then we define constructions and, finally, the ramified hierarchy of types.

Definition 1. *(*types of order 1*). Let B be a* base*, where a base is a collection of pair-wise disjoint, non-empty sets. Then:*

(i) Every member of B is an elementary type of order 1 over B*.*
(ii) Let $\alpha, \beta_1, \ldots, \beta_m (m > 0)$ *be types of order 1 over B. Then the collection* $(\alpha\beta_1 \ldots \beta_m)$ *of all m-ary partial mappings from* $\beta_1 \times \ldots \times \beta_m$ *into* α *is a functional* type of order 1 over B*.*
(iii) Nothing is a type of order 1 over B *unless it so follows from (i) and (ii).*

The types *ad* (ii) are *functional* types. They are sets of *partial functions*, i.e., functions that associate every m-tuple of arguments with at most one value. Thus total functions are a special kind of partial functions.

The choice of the base depends on the area and language we happen to be investigating. When investigating purely mathematical language, the base can consist of, e.g., two atomic types; o, the type of truth-values, and ν, the type of natural numbers. For the purposes of natural-language analysis, we are currently assuming the following base of ground types, which is part of the ontological commitments of TIL:

o the set of truth-values $\{\mathrm{T}, \mathrm{F}\}$;
ι the set of individuals (the universe of discourse);
τ the set of real numbers (doubling as discrete times);
ω the set of logically possible worlds (the logical space).

Since *function* rather than relation is a primitive notion of TIL, we model *sets* and *relations* by their characteristic functions. Thus, for example, the set of prime numbers is a function of type $(o\tau)$ that associates any number with **T** or **F** according as the given number is a prime.

Definition 2. *(*intension and extension*)*
(PWS) intensions *are entities of type* $(\beta\omega)$*: mappings from possible worlds to some type* β*. The type* β *is frequently the type of the* chronology *of* α*-objects, i.e., mapping of type* $(\alpha\tau)$*. Thus* α*-intensions are frequently functions of type* $((\alpha\tau)\omega)$*, abbreviated as '*$\alpha_{\tau\omega}$*'.* Extensions *are entities of a type* α *where* $\alpha \neq (\beta\omega)$ *for any type* β*.*

Examples of frequently used intensions are:
propositions (denoted by declarative sentences) are of type $o_{\tau\omega}$; *properties of individuals* (usually denoted by nouns or intransitive verbs like 'is a student', 'walks') of type $(o\iota)_{\tau\omega}$; *binary relations-in-intension* between individuals are of type $(o\iota\iota)_{\tau\omega}$; *individual offices/roles*(usually denoted either by superlatives like 'the highest mountain' or terms with built-in uniqueness, like 'The President of the USA') are of type $\iota_{\tau\omega}$.

Expressions which denote non-constant intensions (i.e. functions that take different values in at least two world-time pairs) are *empirical*. Note that some

extensions involve the set of possible worlds, but not as their domain. For instance, a *set* of propositions is an extensional entity of type $(oo_{\tau\omega})$. On the other hand, a *property* of propositions, like being true in a world w at time t, is an intensional entity of type $(oo_{\tau\omega})_{\tau\omega}$.

Quantifiers $\forall^\alpha$, $\exists^\alpha$ are extensions, viz. type-theoretically polymorphous functions of type $(o(o\alpha))$, for an arbitrary type α, defined as follows. The *universal quantifier* $\forall^\alpha$ is a function that associates a class A of α-elements with **T** if A contains all elements of the type α, otherwise with **F**. The *existential quantifier* $\exists^\alpha$ is a function that associates a class A of α-elements with **T** if A is a non-empty class, otherwise with **F**.

The *singularizer* $Sing^\alpha$ is a partial type-theoretically polymorphic function of type $(\alpha(o\alpha))$ that associates a class C with the only α-element of C if C is a singleton, otherwise the function $Sing^\alpha$ is undefined.

Where A v-constructs a truth-value, i.e. an o-object and x v-constructs an α-object, we will often use the abbreviated notation '$\forall xA$', '$\exists xA$' and 'ιxA' instead of '$[{}^0\forall^\alpha \lambda x A]$', '$[{}^0\exists^\alpha \lambda x A]$', '$[{}^0\iota^\alpha \lambda x A]$', respectively, when no confusion can arise.

Constructions are assigned to expressions as their algorithmically structured, context-invariant meanings. When claiming that constructions are algorithmically structured, we mean the following. A construction C consists of one or more particular steps, or *constituents*, that are to be individually executed in order to execute C. The objects a construction operates on are not constituents of the construction. Just like the constituents of a computer program are its subprograms, so the constituents of a construction are its sub-constructions. Thus on the lowest level of non-constructions, the objects that constructions work on have to be supplied by other (albeit trivial) constructions. The constructions themselves may occur not only as constituents to be executed, but also as objects that still other constructions operate on. Therefore, one should not conflate *using* constructions as constituents of compound constructions and *mentioning* constructions that enter as input/output objects into compound constructions. So we must strictly distinguish between using constructions as constituents and mentioning constructions as objects.

Mentioning is, in principle, achieved by *using* atomic constructions. A construction C is *atomic* if it does not contain any other construction as a used sub-construction (a 'constituent of C') but C. There are two atomic constructions that supply entities (of any type) on which compound constructions operate: *Variables* and *Trivializations*. *Compound* constructions, which consist of other constituents than just themselves, are *Composition* and *Closure*. Composition is the instruction to apply a function to an argument in order to obtain its value (if any) at the argument. It is *improper*, i.e., does not construct anything, if the function is not defined at the argument. Closure is the instruction to construct a function by abstracting over variables in the ordinary manner of λ-calculi. Finally, higher-order constructions can be used once or twice over as constituents of constructions. This is achieved by a fifth and sixth construction called *Execution* and *Double Execution*, respectively.

Definition 3. (construction).

(i) *The variable x is a* construction *that v-constructs an object O of the respective type dependently on a valuation v.*

(ii) Trivialization*: Where X is an object whatsoever (an extension, an intension or a construction),* ^0X *is the construction* Trivialization*. It constructs X without any change.*

(iii) *The* Composition $[X\, Y_1 \ldots Y_m]$ *is the following* construction*. If X v-constructs a function f of type* $(\alpha\beta_1 \ldots \beta_m)$*, and* $Y_1, \ldots, Y_m$ *v-construct entities* $B_1, \ldots, B_m$ *of types* $\beta_1, \ldots, \beta_m$*, respectively, then the* Composition $[X\, Y_1 \ldots Y_m]$ *v-*constructs *the value (an entity, if any, of type α) of f on the tuple argument* $\langle B_1, \ldots, B_m\rangle$*. Otherwise the* Composition $[X\, Y_1 \ldots Y_m]$ *does not v-construct anything and so is v-improper.*

(iv) *The Closure* $[\lambda x_1 \ldots x_m\, Y]$ *is the following* construction*. Let* x_1*,* x_2*, …,* x_m *be pair-wise distinct variables v-constructing entities of types* $\beta_1, \ldots, \beta_m$ *and Y a construction v-constructing an α-entity. Then* $[\lambda x_1 \ldots x_m\, Y]$ *is the construction* λ-Closure *(or* Closure*). It v-constructs the following function f of the type* $(\alpha\beta_1 \ldots \beta_m)$*. Let* $v(B_1/x_1, \ldots, B_m/x_m)$ *be a valuation identical with v at least up to assigning objects* $B_1/\beta_1, \ldots, B_m/\beta_m$ *to variables* $x_1, \ldots, x_m$*. If Y is* $v(B_1/x_1, \ldots, B_m/x_m)$*-improper (see iii), then f is undefined on* $\langle B_1, \ldots, B_m\rangle$*. Otherwise the value of f on* $\langle B_1, \ldots, B_m\rangle$ *is the α-entity* $v(B_1/x_1, \ldots, B_m/x_m)$*-constructed by Y.*

(v) *The* Single Execution ^1X *is the* construction *that v-constructs the entity v-constructed by X. Otherwise* ^1X *is v-improper.*

(vi) *The* Double Execution ^2X *is the following* construction*. If X v-construct a construction Y and Y v-construct an entity Z, then* ^2X *v-constructs Z. Otherwise* ^2X *is v-improper.*

(vii) *Nothing is a* construction*, unless it so follows from (i) through (vi).*

Notation and abbreviations.
– 'X/α' means that the object X is (a member) of type α;
– '$X \rightarrow_v \alpha$' means that the type of the object v-constructed by X is α. We use '$X \rightarrow \alpha$' if what is v-constructed does not depend on a valuation v;
– We will standardly use the variables $w \rightarrow_v \omega$ and $t \rightarrow_v \tau$;
– If $C \rightarrow_v \alpha_{\tau\omega}$, the frequently used Composition $[[Cw]t]$, the intensional descent of the α-intension v-constructed by C, will be written as 'C_{wt}';
– When using constructions of truth-value functions, namely $\wedge$ (conjunction), $\vee$ (disjunction) and $\supset$ (implication) of type (ooo), and $\neg$ (negation) of type (oo), we often omit Trivialisation and use infix notion;
– When using identity relations $=^\alpha$ $/(o\alpha\alpha)$, we often omit the superscript α and use infix notation, whenever no confusion arises.

As mentioned above, constructions themselves are objects and thus also receive a type. Only it cannot be a type of order 1, because a construction cannot be of the same type as the object it constructs. Constructions that construct entities of order 1 are *constructions of order 1*. They belong to a type of order 2, denoted by '$*_1$'. This type $*_1$, together with atomic types of order 1, serves as the base for the following induction rule: any collection of partial mappings,

type $(\alpha\beta_1 \ldots \beta_n)$, involving $*_1$ in their domain or range is a *type of order 2*. Constructions belonging to the type $*_2$, which identify entities of order 1 or 2, and partial mappings involving such constructions, belong to a *type of order 3*; and so on *ad infinitum*.

The definition of the ramified hierarchy of types decomposes into three parts. First, simple types of order 1 were already defined by Definition 1. Second, we define constructions of order n, and third, types of order $n + 1$.

Definition 4. *(*ramified hierarchy of types*).*

$\mathbf{T}_1$ (types of order 1). *See Definition 1.*

$\mathbf{C}_n$ (constructions of order n).

i) *Let x be a variable ranging over a type of order n. Then x is a* construction of order n over B.

ii) *Let X be a member of a type of order n. Then* ${}^0X$, ${}^1X$, ${}^2X$ *are* constructions of order n over B.

iii) *Let* X, X_1,..., X_m $(m > 0)$ *be constructions of order n over B. Then* $[X\, X_1 \ldots X_m]$ *is a* construction of order n over B.

iv) *Let* $x_1, \ldots, x_m$, X $(m > 0)$ *be constructions of order n over B. Then* $[\lambda x_1 \ldots x_m\, X]$ *is a* construction of order n over B.

v) *Nothing is a* construction of order n over B *unless it so follows from* C_n *(i)-(iv).*

$\mathbf{T}_{n+1}$ (types of order $n + 1$). *Let* $*_n$ *be the collection of all constructions of order n over B. Then*

i) $*_n$ *and every type of order n are* types of order $n + 1$.

ii) *If* $0 < m$ *and* $\alpha, \beta_1, \ldots, \beta_m$ *are types of order* $n + 1$ *over B, then* $(\alpha\, \beta_1 \ldots \beta_m)$ *is a* type of order $n + 1$ over B.

iii) *Nothing is a* type of order $n + 1$ over B *unless it so follows from* T_{n+1} *(i) and (ii).*

So much for the logical machinery we are going to apply in the next paragraphs in order to specify the process-ontology content.

3 Ontology Content

Formal ontology is a result of the conceptualization of a given domain. Typically, a formal ontology encompasses these parts:

(1) Conceptual (terminological) dictionary which contains:
   a) primitive concepts
   b) compound concepts (ontological definitions of entities)
   c) the most important descriptive attributes, in particular identification of entities

(2) Relations
   a) contingent empirical relations between entities, in particular the part-whole relation
   b) analytical relations between intensions, i.e., requisites and essence, which give rise to ISA hierarchy

(3) Integrity constraints
   a) analytically necessary rules
   b) nomologically necessary rules
   c) common rules of 'necessity by convention'

The process of an ontology design usually begins with the specification of primitive concepts, i.e., Trivializations of objects that are not constructions. These primitive concepts are supposed to be commonly understood and they are not further refined. For instance, primitive concepts of traffic-system ontology might be ${}^0Agent$, ${}^0Lane$, ${}^0Crossroads$, etc. Next we specify compound concepts as ontological definitions of entities of a given domain. For instance, a road can be defined as consisting of two or more lanes which pass from a crossroad to another crossroad.

When specifying relations between entities, we distinguish between empirical and analytical relations. The former are relations-in-intension, mostly between individuals, like the part-whole relation. Analytical relations are relations-in-extension between intensions, for instance, a requisite relation and a property typical for another property. These relations give rise to ISA taxonomies. For instance, that a driver is a person is analytically necessary proposition *TRUE* that takes the value **T** in all $\langle w, t\rangle$-pairs. The requisite relation of the type $(o(o\iota)_{\tau\omega}(o\iota)_{\tau\omega})$ between the property of being a driver and the property of being a person is defined as follows:

$$[{}^0Req\ {}^0Person\ {}^0Driver] =_{df} \forall w \forall t[\forall x[[{}^0Driver_{wt}\ x] \supset [{}^0Person_{wt}\ x]]]$$

*Gloss.*Being a person is a requisite of being a driver. In other words, necessarily for any individual x, if x instantiates the property of being a driver then x also instantiates the property of being a person.

On the other hand, a driver typically owns a car, unless he is a chauffeur or a chauffeuse working for somebody else without owning a car. We say that owning a car is a typical property of a driver:

$$[{}^0Typically\ {}^0OwnCar\ {}^0Driver\ {}^0Exception] =_{df}$$
$$\forall w \forall t[\forall x\ [\neg[{}^0Exception_{wt}\ x] \supset [[{}^0Driver_{wt}\ x] \supset [{}^0OwnCar_{wt}\ x]]]]$$

Requisites and typical properties can obtain between intensions of any type. Here we define these relations only between properties of individuals. The other kinds can be easily deduced from this one. Let $p, q, exc \rightarrow_v (o\iota)_{\tau\omega}$; $x \rightarrow_v \iota$; $True/(oo_{\tau\omega})_{\tau\omega}$: the property of a proposition of being true in a world w at time t. Then q is a *requisite* of p, if and only if

$$\forall w \forall t[\forall x[[{}^0True_{wt}\ \lambda w \lambda t[p_{wt}\ x]] \supset [{}^0True_{wt}\ \lambda w \lambda t[q_{wt}\ x]]]]$$

The relation of being a *typical property* is defined as follows:

$$\forall w \forall t[\forall x[\neg[{}^0True_{wt}\ \lambda w \lambda t[exc_{wt}\ x]] \supset$$
$$[[{}^0True_{wt}\ \lambda w \lambda t[p_{wt}\ x]] \supset [{}^0True_{wt}\ \lambda w \lambda t[q_{wt}\ x]]]]]$$

Gloss. p is typical of q unless *exc*(eption).

Note. Due to partiality, we must use the property of propositions of being true. It returns the value **T** if the given proposition takes the value **T** in a $\langle w, t\rangle$-pair, otherwise **F**. If we did not apply this property, then it might be the case that $[p_{wt}x]$ would be v-improper and thus the whole Composition $[[^0True_{wt}\ \lambda w\lambda t[p_{wt}\ x]]\ \supset\ [^0True_{wt}\ \lambda w\lambda t[q_{wt}\ x]]]$ would be v-improper as well, which means that the above Closure would construct **F**. This is wrong, for sure, because the relation of being a requisite, providing it is valid, then it is valid in all $\langle w, t\rangle$-pairs.

Example. The requisite of the property of having stopped smoking is the property of previous smoking. Thus for those individuals y who never smoked the Composition $[^0StopedSmoking_{wt}\ y]$ is v-improper, $StopedSmoking/(o\iota)_{\tau\omega}$; $y \rightarrow_v \iota$.

It is a well-known fact that hierarchies of intensions based on requisite relations establish *inheritance* of attributes and possibly also of operations. For instance, a driver in addition to his/her special attributes like having a driving license inherits all the attributes of a person. This is another reason for including such a hierarchy into ontology. This concludes our summary of the logic of intensions. Now we are going to investigate ontology of processes.

4 Ontology of Processes

The specification of processes is driven by the analysis of verbs that denote actions. For instance, the specification of the process of John's driving from Prague to Ostrava is given by the sense of the verb 'to drive' together with its arguments (*who* is driving: the actor, *from where*, *to where*, etc.).

Our analysis makes use of Tichý's (1980) [10], where such verbs are called emphepisodic verbs. They tell us what people *do* rather than what they *are*. Thus while attributive verbs like 'to be happy', 'to be a good pianist' ascribe to people empirical *properties*, John's driving from Prague to Ostrava is not a condition in which John may be. Rather, it is John's *behaviour*, a time consuming *process* consisting of a series of *events*.

Here we are not going to deal with an exact definition of an event and an episode/process. Referring for details to Tichý [10] (1980), we just briefly summarise. Tichý defines an event as a set of basic propositions (possibly with a time-shift) together with a proposition specifying time when the event occurs. Thus the type of an event is $(oo_{\tau\omega})$. Basic propositions are formed by the application of a basic property to an individual. Basic individual properties are the properties corresponding to pre-theoretical features of individuals which together constitute an intensional base of a given language. A series of events constitutes a process (Tichý's *episode*). Each process has assigned a time span when it occurs. Sure, John's driving from Prague to Ostrava on November 17 is another event than his driving on October 11. Moreover, each process has an *actor* who *does* the process. The *Does* relation is of type $(o\iota(o(oo_{\tau\omega})))_{\tau\omega}$, and it is

defined as follows:

$$\lambda w \lambda t[{}^0Does_{wt}\, x\, p] = \lambda w \lambda t \exists e[[p\, e] \wedge [{}^0Actor\, x\, e] \wedge [{}^0Occur_w\, e] \wedge [[{}^0Run\, e]\, t]]$$

Additional types: $x \rightarrow_v \iota$; $p \rightarrow_v (o(oo_{\tau\omega}))$: a process; $e \rightarrow_v (oo_{\tau\omega})$: an event; $Actor/(o\iota(oo_{\tau\omega}))$: the relation between an individual and an event in which this individual is involved;[4] $Occur/(o(oo_{\tau\omega}))_w$; $Run/((o\tau)(oo_{\tau\omega}))$: the function that given an event returns a time interval when the event occurs in w.

Gloss. An individual x does a process p in world w at time t iff there is an event belonging to the process such that the event occurs in world w and runs at time t, and x is an actor of the event.

In addition to the actor and the time span when the process occurs, many other parameters of a process are desirable to be followed. As stated above, the process specification is driven by episodic verbs. In order to determine the other parameters of a process, we make use of the results of linguistic analysis, in particular of verb-valence frames.

In linguistics, verb valence is characterized as the ability of a verb to be linked to other terms of the discourse.[5] This ability concerns the semantic level of a language that is the deep structure of a sentence. Since valence frames correspond to senses of verbs, we can obtain a finer specification of the process/procedure. From the logical point of view, a verb denotes a relation and the valence of the verb determines arity and the types of arguments of the relation. Thus each process can be specified by a verb (*what* is to be done), with the parameters like the agent/actor of the process (*who*), the objects to be operated on, resources, etc.

Referring for details to `http://ufal.mff.cuni.cz/vallex/2.5/doc/`, we quote:

> Within the *Functional Generative Description (FGD)* framework, valence frames in a narrow sense consist only of inner participants (both obligatory and optional) and obligatory free modifications. In VALLEX 2.5, valence frames are enriched with quasi-valence complementations. Moreover, a few non-obligatory free modifications occur in valence frames too, since they are typically related to some verbs (or even to whole classes of them) and not to others. (The other free modifications can occur with the given verb too, but they are not contained in the valence frame as their presence in a sentence is not understood as syntactically conditioned in FGD.)
>
> In VALLEX 2.5, a valence frame is modeled as a sequence of frame slots. Each frame slot corresponds to one (either required or specifically permitted) complementation of the given verb.
>
> Note on terminology: in this text, the term 'complementation' (dependent item) is used in its broad sense, not related to the traditional argument/adjunct (complement/modifier) dichotomy.

[4] Tichý calls this relation '*By*' and defines it as the relation that obtains between an individual a and an event e whenever a exemplifies the basic properties involved in e, that is those properties that generate basic propositions of which e consists of. [5] For details see, e.g., Lotko (2003) [7].

> The following attributes are assigned to each slot: *functor*, list of possible morphemic forms (realization), type of complementation.
> In VALLEX 2.5, functors (labels for 'deep roles'; similar to theta-roles) are used for expressing types of relations between verbs and their complementations. According to FGD, functors are divided into inner participants (actants) and free modifications (this division roughly corresponds to the argument/adjunct dichotomy). In VALLEX 2.5, we also distinguish an additional group of quasi-valence complementations.

Functors that occur in VALLEX 2.5 are devided into three groups, viz. *inner participants*, *Quasi-valence complementations* and *free modifications*. For our purpose, inner participants are the most important. They are as follows:

ACT (actor),
ADDR (addressee),
PAT (patient),
EFF (effect) and
ORIG (origin).

The other functors, if needed, will be explained when used. Consider, for instance, the simple sentence "Mary is sending a message to Tom". We have the process of *Sending* with three obligatory arguments: *Who* does the process (*ACT*: the actor of the process), *Whom* (emphADDR: addressee) and *What* (*PAT*: the message). Hence necessarily, whenever an actor does the process of *Sending* then there is an addressee and a patient of the process. We will say that *ADDR* and *PAT* are the *requisites* of the *Sending process*.

However, the requisite relation has been defined in Section 3 as the relation-in-extension between *intensions*, but a process is defined as a set of events, i.e., an *extension*, and *ADDR* is obviously an individual and *PAT* is a (hyper)proposition. Thus a question arises, in which sense can we say that the requisite relation obtains between a process and *ADDR*, *PAT*, respectively? As explained at the outset of this section, there are basic properties of an actor or other individuals involved in the process. The requisite relation between the *Sending* process and *ADDR* can thus be explicated as follows:

$$[{}^0Req_{pr}\, q\, {}^0Sending] =_{df}$$
$$\forall w \forall t \forall x [[{}^0Does_{wt}\, x\, {}^0Sending] \supset$$
$$\exists e [[{}^0Sending\, e] \wedge \exists p\, y [[e\, p] \wedge [p_{wt}] \supset [q_{wt}\, y]]]]]$$

Types: $Sending/(o(oo_{\tau\omega}))$; $x, y \rightarrow_v \iota$; $e/(oo_{\tau\omega})$; $p \rightarrow_v o_{\tau\omega}$; $q \rightarrow_v (o\iota)_{\tau\omega}$.

In our example the property q would be the property of being an addressee of the message. Similarly for *PAT*.

The last notion we need to introduce into our ontology is the *classification* of processes. Sure, the process of somebody's getting up or singing the Czech anthem is of a different kind than, for instance, the process of having a car crash. Accordingly we can then assign requisites to all the members of a particular class of processes.

As an example we adduce the processes of the class *Accident*. The obligatory attributes (requisites) of each accident are as follows:

- *ACT* – the actor(s) who caused the accident
- *PAT* – those who are involved in the accident
- *TWHEN* – time when the accident happened to occur

The other optional attributes (typical properties) are:

- *LOC* – where the accident occurred
- *RCMP* – damages
- *CAUS* – the cause of the accident
- *EFF* – the effects of the accident
- *BEN (benefactive)* – who is to be compensated

Having specified these requisites and typical properties of an accident, we can then specify ontological rules like

"The actor is responsible for the damage"
"The patient is the one to be compensated"
"The actor caused the accident"
"Typically, the actor is due to cover the accident compensation"
etc.

These rules make it possible to infer consequences of an accident, to calculate the damage, etc. In this way the process ontology can serve as an information resource for reasoning about the process.

5 Conclusion

In this paper we introduced the method of building up an ontology of processes. We applied the method of logical analysis of natural language expressions as provided within TIL, and made an attempt to exploit the results of linguistic analysis as provided by the so-called verb-valence frames.

Yet we wish to say that the results presented here are just the first proposal, and a lot of problems remain open. For instance, we only briefly tackled the problem of the classification of processes, the problem of inferring consequences of a process occurrence, etc. Thus the ontology of processes is still much work in progress.

Acknowledgements

This research has been supported by the Grant Agency of the Czech Republic, projects No. GACR 401/10/0792, 'Temporal aspects of knowledge and information' and 401/09/H007 'Logical Foundations of Semantics', and also by the internal grant agency of FEECS VSB-TU Ostrava – IGA 22/2009 'Modelling, simulation and verification of software processes'.

References

1. Carnap, R.: *Meaning and Necessity*, Chicago University Press (1947).
2. Duží, M., Číhalová, M., Menšík, M: 'Ontology as a logic of intensions'. In *European-Japanese Conference EJC 2010*, A. Heimburger, Y. Kiyoki, T. Tokuda, N. Yoshida (eds.), Jyvaskyla, Finland: University of Jyvaskyla, 9–28 (2010).
3. Duží, M., Materna, P.: 'Concepts and Ontologies'. In *Information Modelling and Knowledge Bases XX*, Y. Kiyoki, T. Tokuda, H. Jaakola, X. Chen and N. Yoshida (eds.), Amsterdam: IOS Press, (2009) 45–64.
4. Duží, M., Jespersen, B., Materna, P.: *Procedural Semantics for Hyperintensional Logic (Foundations and Applications of TIL)*. Berlin/Heidelberg: Springer, series Logic, Epistemology, and the Unity of Science (2010).
5. Frege, G.: 'Über Sinn und Bedeutung', *Zeitschrift für Philosophie und philosophische Kritik*, vol. 100 (1892), 25–50.
6. Hajičová, E.: 'What we are talking about and what we are saying about it'. In *Computational Linguistics and Intelligent Text Processing*, LNCS Springer Berlin/Heidelberg, vol. 4919, 241–262 (2008).
7. Lotko, E.: *Slovník lingvistických termínů pro filology*, Olomouc (2003).
8. Moschovakis, Y.N.: 'Sense and denotation as algorithm and value'. In J. Väänänen and J. Oikkonen (eds.), *Lecture Notes in Logic*, vol. 2, Berlin: Springer, pp. 210–249 (1994).
9. Sowa, J.: *Knowledge Representation.* Logical, Philosophical, and Computational Foundations. Brooks/Cole (2000).
10. Tichý, P.: 'The semantics of episodic verbs', *Theoretical linguistics*, 7, 263–296 (1980). Reprinted in: Tichý (2004, pp. 409–444).
11. Tichý, P.: *The Foundations of Frege's Logic*, Berlin, New York: De Gruyter (1988).
12. Tichý, P.: *Pavel Tichý's Collected Papers in Logic and Philosophy*, Dunedin: University of Otago Press; Prague: Filosofia, Czech Academy of Sciences (2004).

Linking VerbaLex with FrameNet
Case Study for the Indicate Verb Class

Jiří Materna

Centre for Natural Language Processing
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00, Brno, Czech Republic
xmaterna@fi.muni.cz http://nlp.fi.muni.cz

Abstract. The aim of this work is to evaluate possibilities of linking FrameNet frames with Czech verb valency frames from VerbaLex on the class of Indicate verbs. This class is taken from VerbaLex. The motivation cames from the intention to build a FrameNet-like database of semantic frames in Czech.

Key words: VerbaLex, FrameNet, Indicate verb class

1 Introduction

This work is a part of much more complex task, which aims to build a large, domain independent lexicon of semantic frames in Czech, based on the Frame Semantic formalism, similar to the original Berkeley FrameNet [1].

A fundamental assumption of the methodology of building Czech FrameNet is that the most of Berkeley FrameNet frames can be reused for the semantic analysis of Czech. The assumption takes advantage of the nature of frames as coarse-grained semantic classes, which refer to prototypical situations. Nevertheless, the assumption that these situations are applicable across languages should be empirically verified. In general, a sense of a lemma can evoke a FrameNet frame if this sense is able to realize the conceptually necessary components of the frame (its core frame elements). Inversely, the FrameNet frames cannot be applicable to other languages if the sub-categorization properties of lemmas in this language differ significantly from their English translations.

In this work we try to reveal the most significant problems by carrying out manual linkage of all VerbaLex frames from the Indicate verb class with Berkeley FrameNet.

2 Frame Semantics and FrameNet

Frame semantics is an approach to the study of lexical meaning based on the work by Charles J. Fillmore and his collaborators [2]. The central idea of the frame semantics is that word meaning is described in a relation to *semantic frame*,

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 89–94, 2010.

which consists of a target *lexical unit* (pairing of a word with a sense), *frame elements* (its semantic arguments) and relations between them.

FrameNet is a project in which the information about the linked semantic and syntactic properties of English words is extracted from a large electronic text corpora, using both manual and automatic procedures. The information about words and their properties is stored in an electronic lexical database. Possible syntactic realizations of the semantic roles associated with a frame are exemplified in the annotated FrameNet corpus.

2.1 Semantic Frames

A semantic frame is defined as a script-like conceptual structure that describes a particular type of situation, object or event along with its participants and properties [3].

Lexical unit is a pairing of a word with a meaning. Typically, each sense of a polysemous word belongs to a different semantic frame. For example, the *Commerce_sell* frame describes a situation in which a `seller` sells some `goods` to a `buyer`, and is evoked by lexical units such as *auction*, *retail*, *retailer*, *sale*, *sell*, etc. The semantic participants are called Frame Elements.

2.2 Frame Elements

The semantic valencies of a lexical unit are expressed in terms of the kinds of entities that can participate in frames of the type evoked by the lexical unit. The valencies are called *frame elements*. Frame elements (FEs) bear some resemblance to the argument variables used in first-order predicate logic, but have important differences came from the fact that frames are much more complex than logical predicates [4]. In the example above, the frame elements include `Seller`, `Goods`, `Buyer`, etc.

FrameNet distinguishes three types of frame elements – *core* FEs (the presence of such FEs is necessary to satisfy a semantic valence of a given frame), *peripheral* FEs (they are not unique for a given frame and can usually occur in any frame, typically expressions of time, place, manner, purpose, attitude, etc.) and *extra-thematic* FEs (these FEs have no direct relation to the situation identified with the frame, but add new information, often showing how the event represented by one frame is a part of an event involving another frame).

3 VerbaLex

VerbaLex is an electronic database of verb valency frames in Czech, which has been developed in the Centre for Natural Language Processing at the Faculty of Informatics MU recently. Entries in VerbaLex are formed by lemmata in synonymic relations followed by their sense numbers in standard Princeton WordNet notation. Verb valencies are realized on two levels – deep valency

which corresponds to the semantic role (or selectional restrictions) and surface layer reflecting information about syntactic and morphological valencies.

The current version of VerbaLex contains more than 6,000 synsets, more than 21,000 verb senses and about 10,500 verb lemmata in 19,500 valency frames.

3.1 Complex Valency Frames

The valency frame represents verb valencies on both syntactic and semantic level. In the centre of the complex frame, there is a symbol representing the verb position, surrounded by the left-hand and right-hand arguments in the canonical word-order. The type of valency relation for each constituent element is marked as obligatory or optional. Semantic information about the verbal complement is represented by two-level semantic roles.

The first level contains semantic roles mainly based on the EuroWordNet [5] first-order and second-order top ontology entities, arranged in a hierarchical structure. The list of first level semantic entities is closed and currently contains 33 concepts. On the second level, selected lexical units from the set of EuroWordNet base concepts with relevant sense numbers are used. The list of second level semantic roles is open and currently contains about 1,200 literals.

The complex valency frame comprises basic valency frame and other additional information about verbs. The additional information includes:

- definition of verb meaning
- verb ability to create passive form
- number of meanings for homonymous verbs
- semantic class a verb belongs to
- aspect (perfective or imperfective)
- example of verb use
- types of reflexivity for reflexive verbs

4 Case study for the *Indicate* verb class

In order to discover main typological differences between Berkeley FrameNet frames and Czech valency frames in VerbaLex we have selected VerbaLex frames belonging to the *Indicate* class and carried out their complete linkage to FrameNet frames. The *Indicate* class is one of 111 semantic classes defined in VerbaLex and consists of 136 verb senses in 27 CZWN synsets evoking 119 valency frames.

4.1 Annotation Process

The annotator proceeds one VerbaLex frame at a time and is asked to assign at most one FrameNet frame to it. If the annotator is not able to find appropriate FrameNet frame, the VerbaLex frame will not be annotated and a new Czech FrameNet frame should be defined in future work. There are at least two reasons of the necessity to define a completely new frame [6]:

1. Inadequacy of frame definitions in the corresponding semantic domain or area.
2. Insufficient coverage of the domain in Berkeley FrameNet (i.e. English lexical units and corresponding frames have not been defined yet).

If it is possible to find an appropriate frame from FrameNet, the process of annotation continues with linking semantic roles from VerbaLex with frame elements from FrameNet. At most one frame element can be chosen for a semantic role and no more than one semantic role can be linked to a FrameNet frame element.

If the appropriate FrameNet frame element for a semantic role from VerbaLex frame does not exist, the semantic role should be connected to a newly-defined frame element in future. Nevertheless, the VerbaLex specific addition of frame elements to a FrameNet frame results in a different and more restricted frame. This more specific non-English frame could be related to the English one by a cross-lingual Inheritance relation, whereby it would become a cross-lingual Child frame of the English frame [6].

4.2 Statistics and Typological Divergences

In our experience, the most of VerbaLex frames can be directly linked to a semantic frame from FrameNet, nevertheless, some of the VerbaLex frames require an adaptation or creating a completely new FrameNet frames. In Table 1 there is a list of the most frequently assigned FrameNet frames with numbers of corresponding VerbaLex frames.

Table 1. Assigned FrameNet frames.

FrameNet frame name	VerbaLex frames
`Telling`	30
`Reasoning`	22
`Reveal_secret`	14
`Gesture`	11
`Expressing_publicly`	7
`Forgiveness`	5
`Communication`	4
`Sign`	3
Others	9
None	9

During the annotation phase we have identified 15 FrameNet frames belonging to 119 VerbaLex frames. It means that approximately 8 VerbaLex frames are linked to one FrameNet frame. Among all 119 VerbaLex frames there are 9 frames that cannot be linked to any known frame from FrameNet. For these frames a new FrameNet frame will need to be defined. The coverage of the Indicate class by FrameNet frames is more than 99%.

When evaluating linkage between semantic roles from VerbaLex and frame elements from FrameNet, we have found 19 frames where a new frame element has to be added, two or more frame elements have to be put together, or some frame elements have to be restructuralized. It has at least three reasons:

1. The corresponding frame element is missing.
2. The frame element is too general and has to be divided into more specific ones.
3. The frame element is too specific and has to be replaced by a more general one.

An example of the missing frame element case can be a sentence

Jeho pohled nám naznačil, že nemluví pravdu.
(His look signed us that he is not telling the truth.),

which corresponds to VerbaLex frame

```
AG(co1;<quality:1>)VERB PAT(komu3;<person:1>) INFO(const;<info:1>)
```

and FrameNet Frame `Sign`. This FrameNet frame allows frame elements `Indicated` (`INFO`), `Indicator` (`AG`) and `Degree` but does not allow any FE, which could be connected to the patient (`PAT`).

An example of case 2, where a frame element has to be divided can be a sentence

Ta zpěvačka demonstrovala svou lásku ke zvířatům.
(The singer demonstrated her love of animals.),

which corresponds to VerbaLex frame

```
AG(kdo1;<person:1>) VERB ACT(co4;<act:2>) PAT(komu3;<animal:1>)
```

and FrameNet frame `Expressing_publicly`. This frame allows core frame elements `Communicator` (`AG`), `Content` and `Medium`. In order to connect this FrameNet frame to the VerbaLex frame, we would have to split `Content` FE into two parts or join `ACT` and `PAT` arguments of the VerbaLex frame together.

The last example illustrates case 3, where a frame element is too specific and should be replace by a more general one

Ten pes cenil zuby na kočku za stromem.
(The dog showed the teeth to the cat behind the tree.),

which corresponds to VerbaLex frame

```
AG<animal:1> VERB DPHR<zuby> PAT <person:1|animal:1>
```

and FrameNet frame Gesture. This frame allows core frame elements Addressee (PAT?), Body_part (DPHR), Communicator (AG), Indicated_entity and Message. The problem is in the definition of frame element Addressee, which says 'This is the person to whom a non-verbal Message is communicated', therefore, an animal is not allowed.

5 Conclusions

The presented work describes an evaluation of linking possibilities between Czech verb valency lexicon VerbaLex and FrameNet on the domain of Indicate verb class. The class consists of 136 verb senses in 27 CZWN synsets evoking 119 valency frames. The results showed that the coverage of the Indicate class by the FrameNet frames is more than 99% and more than 82% of linkable FrameNet frames can remain without any modifications of their frame elements.

For the future work, the goal is to build a core of Czech FrameNet based on a complete linkage of VerbaLex to FrameNet. Such FrameNet based lexicon can be used for information retrieval and searching semantic relations in texts. Also other challenging tasks come into consideration, namely in the area of the Semantic Web.

Acknowledgements

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536 and in the National Research Programme II project 2C06009.

References

1. Ruppenhofer, J., Ellsworth, M., Petruck, M.R.L., Johnson, C.R., Scheffczyk, J.: FrameNet II: Extended Theory and Practice (2006) http://www.icsi.berkeley.edu/framenet.
2. Fillmore, C.J.: Frame Semantics and the Nature of Language. In: Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech. Volume 280. (1976) 20–32.
3. Fillmore, C.J.: Frame Semantics. In: Linguistics in the Morning Calm, Seoul, Hanshin Publishing Co. (1982) 111–137.
4. Fillmore, C.J., Johnson, C.R., Petruck, M.R.L.: Background to FrameNet. In: International Journal of Lexicography, Oxford University Press (2003) 235–250.
5. Vossen, P., Hirst, G.: EuroWordNet: A Multilingual Database with Lexical Semantic. Kluwer Academic Publisher (1998).
6. Lönneker-Rodman, B.: Multilinguality and FrameNet. Technical report, International Computer Science Institute, Berkeley (2007).

Part IV

Language Applications

Effective Creation of Self-Referencing Citation Records
System SelfBib

Tomáš Čapek and Petr Sojka

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
xcapek1@aurora.fi.muni.cz, sojka@fi.muni.cz

Abstract. Acquiring citation records from online resources has become a popular approach to building a bibliography for one's publication. LaTeX document preparation system is the most popular platform for typesetting publications in academia. It uses BibTeX as a tool used to describe and process lists of references. In this article we present a simple method that allows the automatic creation of a full self-referencing citation record for a collection of papers typeset and published within one proceedings of a conference. This greatly facilitates access to the bibliography entries for anyone who wishes to use them as part of their own publication.

1 Introduction

Mathematicians, engineers, philosophers, lawyers, linguists, economists and other scholars all appreciate quick access to other people's research not only in terms of its actual content but also to get bibliography entries, and especially if they are using LaTeX and BibTeX for typesetting. With the growth of widely accessible citation databases and search engines that aggregate scholarly literature there is a need to not only retrieve the information it contains but also to provide information about the publications we produce. For example, the automatic parsing of publications provided by Google Scholar does not always correctly identify all necessary bibliographic data and sometimes even mixes different fields up. The same holds for citation extraction services like that offered by Mendeley.[1] To prevent incorrect metadata records, it is the responsibility of each author, editor or publisher, depending on the scope of the publication, to present their own online publications in such a way that avoids the need for guessing on the part of the indexing engine. There are already established channels for the big players (CrossRef, Google Scholar, Elsevier, Springer, Thompson Reuters) to exchange, validate and match paper metadata. Metadata are often retyped, or produced semi-automatically, which is still error-prone. The optimum point is when the metadata are *generated* during the preparation and typesetting of the publication. In this setup, with

[1] http://www.mendeley.com/bibliography-maker-database-generator/

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 97–102, 2010.

batch typesetting systems, such as LaTeX, metadata which appear in the final version of the publication, end up in a metadata record without any human interference. This idea is most likely already employed in commercial systems such as the one by Elsevier and others [1], but we are not aware of any "poor man's solution" for authors and editors. Based on our experience preparing more than twenty multi-author books and proceedings, we have designed and implemented the system *SelfBib* that automates the production of metadata records as a by-product of typesetting multi-author volumes and proceedings. Accurate and timely accessible citation records help to better identify paper duplicates appearing on the Internet, increase the ease of citation and to a degree also the citation rate.

In our view, the best practise is to typeset a book or a proceedings in a single run with a single LaTeX source file via a set of utility scripts. We describe the main aspects of this approach in Section 2. We show how to easily enhance the typesetting process work flow to provide a full and accurate self-referencing citation record with *SelfBib* in Section 3. Finally, we evaluate the "*SelfBib* approach" and its application in Section 4 and wrap up in Section 5.

2 Prerequisities for Typesetting

The main task of a proceedings' editor is to collect and unify the heterogenous papers contributed by authors. More often than not, authoring instructions even allows the use of different systems (Word or TeX) which makes enforcing the publisher's format a very tedious and time-consuming task. In a research setup, it is often expected that editors also provide a table of contents, author or subject index. This could be hardly achieved automatically without typesetting the whole volume in a single (LaTeX) run – otherwise it implies a lot of manual work with any last minute edit. A prerequisite for automated processing of a complete volume is having all contributions converted (or at least their metadata) into a uniform format, or having the metadata collected into one place. Some supporting systems, such as A. Voronkov's *easychair* do provide rudimentary support for editing the Table of Contents pages, but this does not produce a reliable product, especially when working under the pressure of deadlines.

We recommend working with the uniform format, LaTeX, as it is stable, reliable, and widely used by the scientific community. Most metadata are already tagged in the primary source files (`\title`, `\author`) and others are available during the typesetting (e.g. page numbers). The plain (non-binary) format of LaTeX also allows a high degree of automation, and the unification into one format greatly increases the uniformity of the typeset volume. Good and consistent markup then allows many innovative uses, generating multiple indexes (author, name, subject), hypertext linking across the volume and multiple output formats [2], features usually available for monographs only.

We have designed and implemented a system that allows the typesetting of individual articles and the whole volume in one LaTeX run, in parallel from the same files. During the LaTeX run, additional information is written by standard

and custom macros into an auxiliary (.aux) file. This information is sufficient to build full metadata records for the contributed papers and the whole volume. A script is then run on an auxiliary file, which parses and processes the data into the required formats such as BibTeX (see Section 3).

A typical work flow starts with papers being typeset individually, and then the source completeness is checked. Papers are assigned reference numbers, usually by the supporting reviewing system, and files are renamed using a naming scheme based on these unique paper reference numbers. The reference number is used for multiple purposes, e.g. for naming the directory of the paper, for the name of the root paper's TEX source, for prefixing label names in the paper so that they are unique across the whole volume, etc. This naming scheme allows the editing of the tree of LATEX source files to be partially automated. Several scripts have been developed to facilitate the editing process.

The metadata record of publication item contains data of three kinds:

- data provided by authors (title, list and order of authors and their affiliations, abstract,...)
- data supplied by publishers (publisher name, publishing date, ISBN,...)
- data created during typesetting (page numbers)

The author metadata are already tagged in the primary sources, and can be grabbed from there. The publisher's metadata are usually the last items to be typeset, and with good typesetting conventions they are also defined and tagged unambiguously in the LATEX source files of the publication. The idea is to collect all these data during the final LATEX run and create the full metadata records automatically, as a by-product of the volume production.

On the TEX level, the system consists of

- macros for writing the metadata information into an auxiliary file.
- macros and methodology (naming, tagging, placing local macros) to allow the same files to be used when typesetting a single paper or the whole volume.
- scripting automation (Makefile) to manage the series of typesetting actions and calling the appropriate programs in the right order.

3 SelfBib

SelfBib system consists of several components. The main one is a script (implemented in Ruby programming language [3]), which parses the auxiliary (`.aux`) file from a LATEX run of the whole book and produces well-formed `.bib` file where for each paper within the proceedings the metadata about its title, authors, and first and last pages of the paper in the book are retrieved. In addition to that, a cross-reference key to the primary bibliography entry, which contains information common to all of the papers in the book, is added as well.

In Figure 1 is a sample of the *SelfBib* output consisting of the primary entry and one additional entry for a paper.

```
@proceedings{tsd10conference,
title={{Proceedings of the 13th International
        Conference on Text, Speech and Dialogue---TSD 2010}},
year=2010,
editor={Petr Sojka, Ale{\v s} Hor{\'a}k,
        Ivan Kope{\v c}ek and Karel Pala},
nddress={Brno, Czech Republic},
month=Sep,
publisher={Springer-Verlag},
}

@inproceedings{tsd10conference:100,
    title={{Parsing and Real-World Applications}},
    author={John Carroll},
    pages={2--4},
    crossref={tsd10conference},
}
```

Fig. 1. Sample of *SelfBib* output.

SelfBib has several useful features. For maximum portability, all strings are encoded in 7-bit ASCII so that all entries can be copied as they are, regardless of the language the recipient uses for typesetting. All non-ASCII characters are encoded in LaTeX macros. Also, to ensure that all entries sort correctly, the non-ASCII characters use extended syntax delimited by curly brackets as follows: `{<macro><character>}`. For example, the "š" character is encoded `{\v{s}}`. All frequent variants of accented characters are stored separately in a hash structure and can be extended at will. As an alternative output, *SelfBib* can also provide Google Scholar-compliant HTML meta tags[2] instead of BibTeX entries. The meta tags are useful to include in HTML pages which are dedicated to a single paper. As a result, Google Scholar will always index the metadata as they appeared in the paper without guessing and parsing them from the PDF. This increases the citation matching and lining precision and ensures providing correct bibliography entries.

4 Deployment and Evaluation

When we finish the typesetting of a conference proceedings, there is a variety of ways to promote the self-referencing list of citations for it to be as accessible as possible for anyone who might wish to use one or more entries in their own publication. The most straightforward and natural way is to provide the full reference list for download on the conference homepage. This, however, might not be helpful to users who are unaware of the conference itself and are interested in one particular paper in it, which they have found via a search

[2] `http://scholar.google.com/intl/en/scholar/inclusion.html`

engine. For example, for a paper to appear in Google Scholar results,[3] it needs to be either parsed from the PDF or be accessible in a single (landing) HTML page. For its bibliography record to be accurate, the landing page needs to contain a special set of HTML meta tags[4] that describe the metadata. *SelfBib* can produce bibliography entries in this format as one of its options.

Another way to make the citation list available online is to add the `.bib` file to a online bibliographic database dedicated to a particular field of study. For the field of computer science, the DBLP database is the largest and the most popular resource of bibliographic information[5]. BibTeX format is among those supported that can be used to quickly make the whole citation list available to a large number of scholars via the BibTeX ingestion driver.

Once the accurate and complete metadata item reaches any of the main bibliography citation providers, it tends to be spread via records exchange and matching in systems such as Google Scholar, Mendeley, Bibsonomy, CiteUlike, DBLP, CiteSeer, Crossref and others.

We have generated bibliographic records for twenty proceedings to demonstrate the usefulness of our approach. They are available at the project's web page `http://nlp.fi.muni.cz/projekty/selfbib/bib/`. The system has been proven useful and it significantly facilitates citing bibliography items correctly and efficiently, which in turn potentially increases the citation rate of the papers.

5 Conclusion

In this paper, we have introduced an easy method to enhance the typesetting process of a multi-author volume or an academic proceedings to provide a full and accurate self-referencing list of citations as its by-product. Although our approach can only be used with LaTeX and BibTeX systems, its main advantage is that it is fully automated and quite easy to set up. Depending on the deployment method, the list of citations can make it much easier for anyone compiling a bibliography for their own publication to get access to properly formatted metadata about our publications, or even to help promote our publications by exposing it to a larger number of potential readers.

Acknowledgements This work has been partially supported by the Ministry of Education of CR within the Center of Basic Research LC536 and by the European Union through its Competitiveness and Innovation Programme (Policy Support Programme, "Open access to scientific information", Grant Agreement No. 250503).

[3] http://scholar.google.com/intl/en/scholar/inclusion.htm [4] Google Scholar supports the following tag sets: Highwire Press tags, Eprints tags, BE Press tags and PRISM tags. [5] Primary URL is located at: `http://www.informatik.uni-trier.de/~ley/db/`. Alternate server with limited search capabilities can be found at: `http://dblp.uni-trier.de/` [4]

References

1. Bazargan, K.: LaTeX to MathML and back: A case study of Elsevier journals. In: Proceedings of Practical TeX 2004, TUG (2004).
2. Sojka, P., Růžička, M.: Single-source publishing in multiple formats for different output devices. TUGboat **29**(1) (2008) 118–124.
3. Flanagan, D., Matsumoto, Y.: The Ruby Programming Language. (2008).
4. Ley, M.: DBLP – Some Lessons Learned. PVLDB **2** (2009) 1493–1500.

Towards Partial Word Sense Disambiguation Tools for Czech

Tomáš Čapek and Pavel Šmerk

Faculty of Informatics, Masaryk University, Brno, Czech Republic
xcapek1@aurora.fi.muni.cz, smerk@fi.muni.cz

Abstract. Complex applications in natural language processing such as syntactic analysis, semantic annotation, machine translation and especially word sense disambiguation consist of several relatively simple independent tasks. Czech, belonging among Slavonic languages with many inflectional features, requires more effort for such tasks, in comparison with other languages. In this article we present two software tools to tackle morphological disambiguation and multi-word expression recognition for Czech in a cost saving and time-efficient way.

1 Introduction

Word sense disambiguation (WSD) is the most fundamental task in NLP. In past decades much effort has been made to develop tools to resolve WSD in its entirety, i.e. to correctly disambiguate all words in all contexts. This issue is discussed in detail in [3] as largely unsuccessful. Numerous authors are cited to discuss the reasons behind this and conclude that:

- Many WSD systems assign sense labels from pre-established lexical resources (sense inventories) such as traditional dictionaries and are therefore relative to the sense inventory used, content of which may be at each instance subject to interpretation and might ultimately be unsuitable for some applications. Quality of sense inventories is however not the focus of this paper.
- Whether a sense inventory is used or not, the focus is too often set on division of senses that is too fine-grained even for a human user to distinguish. The effort to encompass as much exceptions and rare sense occurrences can lead to needless complexity of the WSD system whereas NLP can provide useful results by relying on far less.
- Computational WSD should reorient itself to tasks it can easily perform with high accuracy even if they only provide partial results compared to full WSD.

By combining partial solutions for WSD we can make our results more accurate in overall – in ideal case, each step in the text processing might be able to filter out some of the potential variants of the particular word. Below we present two software tools that allow us to partially disambiguate words or collocations in Czech texts. In Section 2 we introduce *Desamb*, a hybrid morphological tagger and Section 3 deals with multi-word expression recogniser called *mrec*.

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 103–108, 2010.

2 Desamb – Morphological Tagger

Morphological analysis is a process which assigns all possible pairs of a lemma and a morphological tag to an analysed word form. A morphological guesser does the same for words unknown to the morphological analyzer. Morphological disambiguation, also called tagging, is a process to determine which lemma and morphological tag is correct with respect to a particular context of the analysed word form in a sentence.

Desamb is an experimental hybrid tagger for Czech, in which rule-based and statistical algorithms are combined [9]. The disambiguation process consists of several independent tools whose inputs and outputs are managed by additional scripts. These tools include morphological analyser, morphological guesser, chunk parser and a tagger based on hidden Markov models (HMM).

As an input, *Desamb* accepts vertical file where the text is stored in the format of one word form or punctuation token per line. The first step of the disambiguation process is a simple detection of sentence boundaries. Then each word form is assigned all its possible pairs of a lemma and a morphological tag by morphological analyzer *ajka* [6]. Similarly, morphological guesser for Czech then computes the same information for word forms that are not covered by *ajka* dictionary [10]. This step concludes the necessary preprocessing of the input text.

Next phase performs the actual disambiguation and can be divided into two steps. In the first step, various lexical filters and morpho-syntactic rules are applied on the data to remove obviously incorrect tags from the list of potential ones for each word form. The filters can be context-independent, for example pronouns *si* or *mi*, which are very frequent in texts, are also recognized by *ajka* as solmization syllables, i.e. nouns. In real texts there are virtually no occurences of this alternative so it can be omitted altogether without causing any measurable inaccuracy in the results. Other filters use simple context information, for example *Se* at the beginning of a sentence can never be a pronoun but is always a preposition. More complex rules are part of a partial syntactic analyser DIS (also called a chunk parser), which recognizes noun, prepositional, and verb phrases in sentences [11]. In Czech, these phrases regularly demostrate certain agreements among several grammatical categories which allows us to remove such tags that do not correspond to given phrase pattern described by a rule.

These filters and rules are designed to be as accurate as possible even if they perform with low recall. Their ambition is always to make the resulting ambiguity lower and to never remove a correct tag. In the end it is still possible for some word forms to have more than one tag attached to them, so the disambiguation at this stage is only partial. On the other hand, these filters and rules are highly reliable and can also be used independently on other tools discussed in this paper.

Finally, a statistical trigram algorithm is used to prune the remaining tags that still need to be disambiguated. An HMM tagger represents the implementation of this approach. At this step each word form is left with exactly one morphological tag.

Because the guesser cannot deal properly with foreign words, especially names, the overall precision of *Desamb*, i.e. the portion of word forms with the correct morphological tag, is 91.0%. When we exclude the foreign words (or, if we can assume flawless morphological analysis of the input text) the precision increases to 95.3%. These results are however quite preliminary, as they were experimentally computed on just 2000 tokens originating in newspaper articles where names occur relatively frequently.

In overall, the main advantage of our approach is its modularity. The individual components of *Desamb* can be replaced or left out. For example, if we use just the HMM tagger, we get fully disambiguated results with relatively low accuracy while using only filters and rules yield highly accurate but partial disambiguation.

3 mrec – Multi-Word Expression Recogniser

Multi-word expression (MWE) recognition is one of the important tasks in NLP. For many applications we need to process MWEs (collocations) as standalone lexical entities for the purpose of lemmatization or parsing. By *MWE* or *collocation* we understand a lexical unit whose meanings can't be inferred from the meanings of the words that make it up, i.e. set phrases, compound words and idioms, rather than any statistically significant word group occurring in a large volume of texts, such as a corpus. To be more specific, we count the following categories among collocations – all can be inflected in Czech:

- multi-word named entities such as toponyms, geographical, proper and other names (e.g. Mediterranean Sea, Julius Caesar, Creative Commons),
- general collocations and set phrases (e.g. carnivorous plant, elementary school, red tape),
- multi-word abbreviations (e.g. a. s., před n. l.),
- Czech reflexive verbs (e.g. kolíbat se, usnadnit si) — these, along with phrasal verbs constitute vast majority of Czech multi-word expressions among verbs. In English, this majority is represented by phrasal verbs alone (e.g. catch on, take off).

One motivation to develop a MWE recogniser closely relates to WSD. Lexical units intrinsically possess a feature of having exactly one sense if they consist of more than one word as it has been verified by D. Yarowsky in [12]. By exploiting this feature we can basically get a partial disambiguation of any text "for free"[1]

Statistical techniques in MWE recognition provide rather approximate results and are more suitable for discovering general multi-word regularities that are outside the scope of our definition of a collocation. For reliable semantic classification of collocations we prefer to utilize rule-based methods and to have a large MWE database at our disposal.

[1] We can also observe similar feature of lexical units if we consider how many senses of a unit we get within one discourse. It has been verified with high accuracy that it is one sense per discourse [2].

We have developed a large Czech MWE database which at the moment contains 160,470 lexical units. It was compiled mostly semi-automatically from various resources such as encyclopedias and dictionaries, public databases of proper nouns and toponyms, collocations obtained from Czech WordNet [4], botanical and zoological terminologies and others. It was originally built for a question answering system UIO which inevitably influenced its composition [7]. The current data can serve as a metalexicon because for each entry the reference to a real dictionary or similar resource is available – this increases the quality of our data in comparison with its previous version [5].

In Table 1 we present basic statistics of the MWE types in the database and their frequencies in SYN2000 corpus which contains 114,363,813 tokens and is a part of Czech National Corpus (CNC) [1].

- column # *MWEs* contains a numbers of the MWEs in our database for each given domain.
- column # *Occs* presents a number of MWE occurrences of each given domain in the SYN2000.
- # *Unique* is a number of individual MWEs from a given domain which occur in the SYN2000 at least once.
- *% of all* represents percent of MWEs occurring in the SYN2000 in comparison with all MWEs from a given domain.
- # *HL* denotes *hapax legomena*, i.e. MWEs with only one occurrence in the SYN2000.
- # *not in corpus* is a number of the MWEs, which did not occur in the SYN2000.

Table 1. Statistics of Czech MWE Database.

Domain	# MWEs	# Occs	# Unique	% of all	# HL	# not in corpus
Botanics, zoology	63,153	13,707	3,279	5.1	1,538	59,874
Culture	6,828	30,279	2,042	29.9	505	4,786
Toponyms	14,561	102,683	2,554	17.5	652	12,007
Proper names (people)	61,152	289,794	15,092	24.7	3,851	46,060
Unsorted	14,776	656,971	7,628	51.6	774	7,148
Total	160,470	1,093,434	30,595	19.1	7,320	129,875

Mrec itself consists of one script that accepts the output from *Desamb* in a form of vertical text file with each line looking as follows:

```
word_form <l>word_form_lemma <c>morphological_tag
```

The vertical file represents the preprocessed corpus data – word order and sentences are preserved. Due to complex inflection in Czech, each line in the *mrec* MWE database uses the following syntax:

$c_1\ c_2\ \ldots\ c_n\#l_1\ l_2\ \ldots\ l_k\grave{}l_{k+1}\ l_{k+2}\ \ldots\ l_{k+m}$

In this format, variables c_x denote component words of a collocation while variables l_x represent lemmata of the respective component words. Character "`" is used as a mark that divides the collocation in two parts; the component words to the left from the mark may be inflected while other component words to the right from the mark have their word forms fixed. For example, there is a collocation in Czech *běžný účet platební bilance (current account of balance of payments)* which would have the mark right in the middle. Both left and right part of the lemmatized form of the collocation may be empty, i.e. the whole collocation may be either fully inflectable or fully fixed.

The purpose of the mark is to help filter out collocations which have some of its component words incorrectly inflected. It also increases performance of the collocation recognition process as it is unnecessary to check fixed component words for any inflection they could otherwise demonstrate. If a collocation exists as a sub-collocation of another and both are recognized in the database, only the longer is by default returned to output.

Mrec is designed as a lightweight tool to be used as a component in a larger system. High processing speed was a major issue in its development. We can report that *mrec* processes as much as 6,000 input tokens per second.

4 Future Work

In the future work we intend to optimize the performance of *Desamb*, specifically the chunk parser and processing speed of the morpho-syntactic rules. The implementation is done in Prolog programming language which is not optimal for tagging large volumes of corpora texts.

Our primary application for the tools described in this paper is to improve semantic annotation of free text with WordNet database serving as the main sense inventory. One of the long-term tasks to do this is to enrich the WordNet database with collocations from our MWE database that are missing in the semantic network and thus can't be used for the annotation. Another way to improve the results is to decrease the sense granularity in the WordNet lexical data with the help of specially designed heuristic tests [8].

Desamb can also be exploited as a part of text processing during the annotation. In this task only information about part-of-speech is necessary to get about the words on input as every lexical unit in the sense inventory is stored as a lemma. By simply adding *Desamb* to the process we get partial WSD "for free" as obviously incorrect morphological tags get filtered out even before we get to recognize collocations. The primary goal of this approach in general is, at each step, get at least slightly less ambiguous form of input text.

5 Conclusion

We have presented two software tools that provide us with partial disambiguation of Czech texts at two different levels – morphological tags and multi-word

expression recognition. By combining them with other techniques such as discourse boundary spotting or word sense labeling we predict can get reasonable and useful WSD results without developing a standalone, monolithic and complex system dedicated to the problem specifically.

Acknowledgements

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536 and by the Czech Science Foundation under the project 407/07/0679.

References

1. *Český národní korpus – SYN2000*. Institute of the Czech National Corpus, Faculty of Arts, Charles University, Praha, Czech Republic, 2000.
2. W.A. Gale, K.W. Church, and D. Yarowsky. One Sense per Discourse. In *Proceedings of the Workshop on Speech and Natural Language*, pages 233–237. Association for Computational Linguistics, 1992.
3. N. Ide and Y. Wilks. Making Sense About Sense. *Word Sense Disambiguation*, pages 47–73, 2006.
4. K. Pala and P. Smrž. Building Czech Wordnet. *Romanian Journal of Information Science and Technology*, 7:79–88, 2004.
5. K. Pala, L. Svoboda, and P. Šmerk. Czech MWE Database. In *Proceedings of the Sixth International Language Resources and Evaluation Conference (LREC'08)*, pages 1–5. European Language Resources Association (ELRA), 2008.
6. R. Sedláček. *Morphemic Analyser for Czech*. Ph.D. thesis, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 2005.
7. L. Svoboda. Processing of Natural Language Multiword Expressions. In V. Snášel, editor, *Proceedings of Znalosti 2004*, Ostrava, Czech Republic, 2004. Technical University Ostrava.
8. T. Čapek. Semantic Network Integrity Maintenance via Heuristic Semi-Automatic Tests. In *Proceedings of the RASLAN Workshop 2009*, pages 63–67. Masaryk University, Brno, Czech Republic, 2009.
9. P. Šmerk. *K morfologické desambiguaci češtiny (On Morphological Desambiguation of Czech)*. Ph.D. thesis proposal, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 2007.
10. P. Šmerk. Towards Czech Morphological Guesser. In P. Sojka and A. Horák, editors, *Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2008*, Brno, 2008. Masarykova univerzita.
11. E. Žáčková. *Parciální syntaktická analýza češtiny (Partial Syntactic Analysis of Czech)*. Ph.D. thesis, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 2002.
12. D. Yarowsky. One Sense per Collocation. In *Proceedings of the workshop on Human Language Technology*, pages 266–271. Association for Computational Linguistics, 1993.

Acquiring NLP Data by means of Games

Marek Grác and Zuzana Nevěřilová

NLP Centre, Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic

Abstract. For some of the NLP tasks, obtaining appropriate data is very difficult. In this paper we concentrate on acquiring NLP data by means of games. Two different projects are presented and various aspects of these games are discussed.
First, we discuss public-made collections of linguistic data generally: the quality and reliability of contributors and collections, and the high dependence of number of contributions on motivation and contribution policy. Second, we describe creation of games for acquiring NLP data in detail. As example, two existing games are presented. Finally, evaluation techniques for both projects are discussed.

1 Introduction

Language resources suitable for natural language processing are one of the keys to develop a successful project. Data of higher quality necessary for some tasks tends to be obtained only with difficulty and a lot of time and manpower is needed. Using large corpora can help, but still data has to be verified by a human. There are two basic approaches on how to collect and/or verify this data. Work can be done by experts or by non-experts (usually volunteers). Both approaches and many in-between variants differ in several aspects such as cost, quality and coverage.

This article focuses on basic aspects of obtaining data from non-expert volunteers. According to [1], we expect to acquire plausible data with lower costs than in the case of expert annotators. We test this expectation on two different projects that both use game as a tool for obtaining data.

In Section 2 we introduce data collections made by general public and discuss the advantages and disadvantages of this approach. We formulate the terms under which a public-made collection can be useful. In Section 3 we present two games that were designed to collect linguistic data. We discuss common aspects and differences of the games. In Section 4 we outline the evaluation of collected data.

2 Public-Made Collections

Internet proves to be very useful tool for grouping people willing to help. They do not need to be at same place, in same timezone or even be willing to

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 109–114, 2010.

gather at same time. Research of crowdsourcing (crowd + outsourcing) became very popular in psychology and sociology. Crowdsourcing is still something that cannot be well defined but we can present some of the advantages and disadvantages.

One of the main advantages is that we can work with people who do not belong to one specific social group (e.g. academic) and so we can receive various views on same problem. Also we can speak to people who are not interested in helping continuously, but they just wish to correct some of our data.

The main disadvantage of this approach is the fact that contributors' expertise may differ a lot. Also we are not able to focus the crowd to work on a schedule or on subsets that are in the worst shape. Crowd will work on what it believes is best for it or tries to offer something better. There are several projects based on content delivered by volunteer work: Wikipedia[1], OpenMind Initiative [2], Games with a Purpose (GWAP Portal) [3], several games including Amazon Mechanical Turk [4] or OnlineWord Games for Semantic Data Collection [1].

This concept can bring plausible results if it fulfils the conditions described in the subsections below.

2.1 Motivation for Players

Since we can only consider the data valuable if it is of sufficient volume, we emphasise the motivation for players to play. People will play a game because it is enjoyable, not because it helps computational linguistics.

The motivation for players to play is fun. First, the design of the game is significant. Second, players playing a game have to beat high scores or advance to new levels, which is often a good motivation. In future, we may consider other motivation such as monthly prizes for best players.

2.2 Formulation of the Problem

The game has to be understandable. Although it can be difficult to play, it should not be difficult to understand the rules. Both games presented in this paper are quite difficult to play but take advantage of the fact that they are only slightly modified but well-known games.

Playing a difficult game is also motivating. Among thousands or even millions of web users, it becomes a challenge to get to the high score list.

From the computational linguistics point of view, we need the game rules to correspond to a problem under consideration. We need the highest possible number of contributors to input the 'right' data. Sometimes it is useful to put restrictions on game rules. Since we always work with semantic data we can set up the rules so that the obtained data will be semantically disambiguated.

[1] http://en.wikipedia.org

2.3 Game Policy and Quality of Contributions

There are several measure for quality of contributions depending of the game type. The most used and most straightforward is the agreement of several players.

We have to consider involuntary errors: For example, a time limit can lead to spelling errors. Players compelled by time limit often write the first idea that comes to their mind. For example: the task of *describing a frog* leads to descriptions such as 'frog is a princess' instead of 'frog is an animal that transforms to princess after you kiss it'. This is not necessarily a disadvantage.

We have to accept that not all contributors understood the game well. For this reason a reliability measure is considered useful for every game (and every set of contributors).

Besides involuntary errors, we have to cope with players contributing deliberately faulty inputs. Primarily we encourage players to register. Contributions by registered players are considered more reliable. Registration has to bring benefits such as higher levels or access to game statistics. In case of a large number of players, we have to find automatic or semi-automatic ways to discover 'hostile' contributors and filter out their contributions.

The games have to take into account language specific features. In case of Czech we have to deal with nominal inflection, by integrating a lemmatizer [5] for finding the appropriate basic form or (in case of X-plain) for generating appropriate word form.

Some web users are used to write *without* diacritics, even if they are normally used in Czech. We have observed the collected data for a period of time and decided that such users form a minority and words without diacritics can be disposed.

3 Games

The following subsections describe two existing games that were designed for collecting data for NLP.

They have several aspects in common such as:

- they refer to existing desktop games
- they are difficult to play
- they are extremely difficult to play for non-native speakers

They differ in aspects such as:

- cooperative/competitive approach
- game based on human-computer or human-human interaction
- suitable for occasional/regular players.

3.1 X-Plain

X-plain [6] has analogy in board games or TV Shows. It is significantly inspired by Verbosity [7], but the engine is based upon word sketches provided by Sketch Engine [8]. It is a cooperative game for two players – a human and a computer. The principle is that a random word (called *secret word*) is displayed to one player (narrator) and s/he has to explain it to the second player (guesser). The guesser has to write down the exact word.

In X-plain the game is time-limited to 3 minutes, therefore it is suitable for occasional players. There are different relation types that together with the *secret word* and the *object* make sentence templates, e.g. `X is_kind_of Y`.

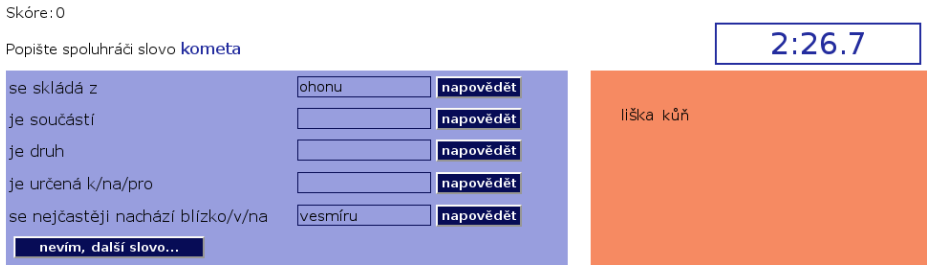


Fig. 1. Screenshot (part) from X-plain: narrator (human) has to describe the word "kometa" (comet). On the left s/he has to fill the following sentence templates: se skládá z (consists of); je součástí (is part of); je druh (is a type of); je určena pro/k/na (is used for); se nejčastěji nachází blízko/v/na (can be likely found). They type: "...se skládá z ohonu (...has part tail). On the right the guesser (computer) tries to guess the secret word: "liška" (fox), "kůň" (horse). There is a countdown timer in the top right corner of the screen.

X-plain is a web-based application and its server side is programmed in PHP, while the client side uses Javascript and AJAX[2] for better user comfort. Contributions are stored in MySQL database.

Figure 1 shows the game interface. When human plays the role of the narrator, his descriptions are stored in form of triples (subject, relation, object) together with their number of occurrences. Triples contain words or word expression in their base forms (lemma), as provided by a lemmatizer [5]. The database is already quite large (nearly 5,000 unique triples in October 2010) and continuously grows. So far the only criterion of contribution quality is frequency: the more often a triple appears, the more probably it is a 'good' one.

3.2 Game of Scrabble

Second game we wish to present is based on the well-known game Scrabble. In this game, players compete against each other to obtain the highest score. They

[2] Asynchronous Javascript and XML

are using letters with different point values and they use the letters to create new word(s) on a gameboard. This game is one of the most popular word games in the world.

There is no problem to play Scrabble on the Internet even for minor languages like Czech. Sites are available where you can meet other players and start a new two-player game. Unlike in X-plain game there is no direct way how to focus player efforts on a specified subset of our problem. Even though the data is more diffused, its quality is much higher because each player's turn has to be confirmed by another player. Some of the word forms are used quite rarely outside of Scrabble world, so developer of the game is extending dictionary of correct words and such words are confirmed automatically. Thus a good dictionary is in players' interest.

Another difference between Scrabble and X-plain is that Scrabble is time-constrained but one game usually takes at least half an hour. There are players who take this game quite seriously and they play more than 150 games per month. If a player plays as many games and they have good winning ratio, we can consider their data to be 'better' then average. It is in our best interest to keep these players interested in our games. To achieve this, we need to offer additional services, even though they do not give us useful data directly—however, they really help to make players more loyal.

We can use scrabble as a tool not only for obtaining new words but also for verifying our existing morphological database. This is not very useful for common words but there are a lot of word forms that are not widely used and some of them are not covered by existing corpora. Good scrabble players tend to know and discuss these forms and their verification can improve our morphological database too.

4 Conclusion and Future Work

This paper describes a different approach to collecting linguistic data. It is designed mainly for collecting linguistic data for Czech language. Czech is a minor language, therefore we cannot expect millions of contributions within a few months like GWAP [3] and have to attend strongly to players' motivation. On the other hand, we can assume only native-speakers will play and no foreigners' language errors will appear.

For each game we record a game history (in case of Scrabble with response latency). Therefore we can identify the pitfalls that players have to face. Further analysis should answer the question why some cases are 'easy' and others are not. We have to carefully choose the data for each level so that players stay motivated.

Beyond these practical questions concerning the games themselves we have to test the resulting collections. We expect that a reasonable number of contributions will be collected over time. We also expect that evaluation techniques to reduce noise in the data will need to be designed in the future.

So far, the data is still not so numerous for a serious evaluation. We plan to evaluate each collection by different means. In case of X-plain, the rising associative network can be compared to other associative networks such as Czech WordNet [9]. Some types of relation are expected to appear in both resources. In case of Scrabble, the evaluation is planned to be manual or semi-automatic and the method of multiple annotation will probably be used.

Acknowledgements

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536.

References

1. Vickrey, D., Bronzan, A., Choi, W., Kumar, A., Turner-Maier, J., Wang, A., Koller, D.: Online word games for semantic data collection. In: EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Morristown, NJ, USA, Association for Computational Linguistics (2008) 533–542.
2. Stork, D.G.: Open mind initiative — about (2007) Retrieved October 28, 2007 from `http://openmind.org`.
3. von Ahn, L.: Games with a purpose. Computer **39**(6) (2006) 92–94.
4. Snow, R., O'Connor, Jurafsky, D., Ng, A.: Cheap and fast—but is it good? evaluating non-expert annotations for natural language tasks. Proceedings of EMNLP-08 (2008).
5. Šmerk, P.: Fast morphological analysis of Czech. In: Proceedings of the Raslan Workshop 2009, Masarykova univerzita (2009).
6. Nevěřilová, Z.: X-plain – a game that collects common sense propositions. In: Proceedings of NLPCS, Funchal, Portugal, SciTePress (2010) 47–52.
7. von Ahn, L., Kedia, M., Blum, M.: Verbosity: a game for collecting common-sense facts. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, ACM (2006) 75–78.
8. Kilgarriff, A., Rychlý, P., Smrž, P., Tugwell, D.: The Sketch engine. In: Proceedings of the Eleventh EURALEX International Congress. (2004) 105–116.
9. Fellbaum, C.: WordNet: An Electronic Lexical Database (Language, Speech, and Communication). The MIT Press (1998).

Author Index

RASLAN 2010

Fourth Workshop on Recent Advances in Slavonic Natural Language Processing

Editors: Petr Sojka, Aleš Horák
Typesetting: Petr Sojka
Cover design: Petr Sojka

Printed and published by Tribun EU s. r. o.
Cejl 32, 602 00 Brno, Czech Republic

First edition at Tribun EU
Brno 2011

ISBN 978-80-7399-246-0

www.librix.eu