Czech Word Sketch Relations with Full Syntax Parser

Aleš Horák[1], Pavel Rychlý[1], and Adam Kilgarriff[2]

[1] Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{hales,pary}@fi.muni.cz
[2] Lexical Computing Ltd.
Brighton, UK
adam@lexmasterclass.com

Abstract. This paper describes the exploitation of dependency relations obtained from syntactic parsing of Czech for building new Czech Word Sketch tables.

Standard Word Sketch construction process usually uses so called Sketch grammars – a simplified process of identifying dependency relations based on regular expressions. This may, of course, lead to errors, which should however not influence (so much) the overall numbers computed on a very big corpus.

The paper presents an experiment of using relations resulting from full syntactic parsing – will they perform better than the standard Sketch grammar or not?

1 Introduction

Dictionary making involves finding the distinctive patterns of usage of words in texts. State-of-the-art corpus query systems can help the lexicographer with this task. They offer great flexibility to search for phrases, collocates, grammatical patterns, to sort concordances to a wide range of criteria and to identify subcorpora for searching only in texts of a particular genre or type. The Sketch Engine [1] is such a corpus query system.

In this paper we discuss the work involved in setting up the Sketch Engine for the new Czech corpus named CZES using two different systems for the dependency relations discovery – the standard Sketch Grammar approach based on regular expressions, and dependency relations obtained by means of full syntax parsing of Czech. We give a detailed description of the various features of the Sketch Engine in relation to the Czech language. The structure of this paper is as follows. First we give some background information on the new CZES corpus and its setup within the Sketch Engine. Then we discuss some general features of the Sketch Engine in Section 3 followed by a detailed description of the work involved in setting up the Sketch Engine for the two sources of dependency relations. We conclude with a short evaluation in the last section.

2 The New Corpus CZES

The Institute of Czech National Corpus has prepared several large Czech corpora. The data of these corpora are provided for research only through web access, it is not possible to add new annotation and process texts by specific batch tools. This is the main reason

Dana Hlaváčková, Aleš Horák, Klára Osolsobě, Pavel Rychlý (Eds.): After Half a Century of Slavonic Natural Language Processing, pp. 101–112, 2009.

why a new Czech Corpus CZES was built in the Masaryk University NLP Centre. CZES was built purely from electronic sources by mostly automated scripts and systems. The corpus name is an acronym of **CZ**ech **E**lectronic **S**ources.

Texts in the CZES corpus come from three different sources:

1. automated harvesting of newspapers (either electronic version of paper ones or electronic only), with annotation of publishing dates, authors and domain; these information is usually hard to find automatically from other sources;
2. customized processing of electronic versions of Czech books available online; and
3. general crawling of the Web.

The whole corpus should contain Czech texts only. There are small parts (paragraphs) in Slovak or English because they are parts of the Czech texts. Some Czech newspapers regularly publish Slovak articles, but we have used an automatic method to identify such articles and remove them from the corpus.

There was no restriction on the publication date of texts. There are both latest articles from current newspapers and 80 year old books present in the corpus.

We are adding more texts to the corpus, the current full corpus size is about 600 million word forms. To speed-up processing and research of different annotations, the work described in this paper uses only a sample of about 85 million tokens from the whole CZES corpus.

In order to support lexicographic searches such as searches by lemma, by part of speech and the extraction of words belonging to a specific word class, the corpus has been annotated with lemma and morphological tags. We have used the Czech tagger DESAMB developed at the NLP Centre [2]. The tagger is based on morphological analyzer AJKA [3] and uses so called “Brno” tag-set for morphological tags.

2.1 Preparing the Corpus

The Sketch Engine input format, often called "vertical" or "word-per-line", is as defined at the University of Stuttgart in the 1990s and widely used in the corpus linguistics community. Each token (e.g. word or punctuation mark) is on a separate line and where there are associated fields of information, typically the lemma and a POS-tag, they are included in tab-separated fields. Structural information, such as document beginnings and ends, sentence and paragraph markup, and meta-information such as the author, title and date of the document, and its text type, are presented in XML-like form on separate lines – see an example from CZES in Figure 1.

A special tag, <g>, was added before punctuation marks: it has the effect of suppressing the space character which is otherwise output between one token and the next. (G is for ’glue’ as the <g> tag ’glues’ the punctuation onto the preceding word.)

The `<s>` tag is used to annotate sentence boundaries and it was added by the tagger.

3 The Sketch Engine

The Sketch Engine is a sophisticated corpus query system. In addition to the standard corpus query functions such as concordances, sorting, filtering, it provides *word sketches*,

```
<doc id="autodesk/1995/05/7" t_main="sci1" medium="cdrom"
    t_orig="Software" lang="cs" title="Autodesk WorkCenter"
    auth_n="Petr Kumprecht" source="CD Modrých stránek"
    d_publ="1995-10" t_sub="inf">
<head>
<s>
Autodesk        Autodesk        kA
WorkCenter      WorkCentra      k1gFnPc2
</s>
</head>
<p>
<s>
Document        Document        k1gInSc1
Management      management      k1gInSc1
a               a               k8xC
Workflow        Workflow        k1gInSc1
Management      management      k1gInSc1
System          System          k1gInSc1
Začátkem        začátkem        k7c2
letošního       letošní         k2eAgInSc2d1
roku            rok             k1gInSc2
uvedla          uvést           k5eAaPmAgFnS
společnost      společnost      k1gFnSc1
Autodesk        Autodesk        kA
na              na              k7c4
trh             trh             k1gInSc4
zcela           zcela           k6eAd1
nový            nový            k2eAgInSc4d1
systém          systém          k1gInSc4
pro             pro             k7c4
správu          správa          k1gFnSc4
dokumentace     dokumentace     k1gFnSc2
<g/>
,               ,               kIx,
Autodesk        Autodesk        kA
WorkCenter      WorkCenter      k1gInSc1
<g/>
.       .       kIx.
</s>
```

Fig. 1. An example of the corpus vertical format with document meta-data.

one page summaries of a word's grammatical and collocational behaviour by integrating grammatical analysis.[3]

Based on the grammatical analysis, the Sketch Engine also produces a distributional *thesaurus* for the language, in which words occurring in similar settings, sharing the same collocates, are put together, and *sketch differences*, which specify similarities and

[3] The Sketch Engine prefers input which has already been lemmatized and POS tagged. If no lemmatized input is available it is possible to apply the Sketch Engine to word forms which, while not optimal, will still be a useful lexicographic tool.

differences between near-synonyms. The system is implemented in C++ and Python and designed for use over the web.

Once the corpus is loaded into the Sketch Engine, the concordance functions are available. The lexicographer can immediately use the search boxes provided, searching, for example, for a lemma specifying its part of speech. This search is case-sensitive as generally lemmas starting with uppercase need to be distinguished from those starting with lower case.

We must note here that the quality of the output of the system depends heavily on the input, i.e. the quality of tagging and lemmatization, which as mentioned in Section 2 is not always entirely satisfactory. According to the sources of some parts of the CZES corpus, the texts can contain misspelled words and neologism, which are tagged by the *guesser* module of the tagger.

On the results page the concordances are shown using KWIC view. With VIEW options it is possible to change the concordance view to a number of alternative views. One is to view additional attributes such as POS tags or lemma alongside each word. This can be useful for finding out why an unexpected corpus line has matched a query, as the cause could be an incorrect POS-tag or lemmatization.

It is central to the process of corpus lexicography that lexicographers often want to insert example sentences from the corpus into the dictionary. Some corpus sentences make good dictionary examples, but others do not. Perhaps they are too long, or too short, or are not well-formed sentences, or contain obscure words or spelling mistakes or abbreviations or strange characters. To find a good dictionary example is a high-level lexicographic skill. But to rule out lots of bad sentences is easy, and the computer can help by doing this groundwork. A new function, GDEX (Good Dictionary Example eXtractor [4]) was added to the Sketch Engine in 2008. This takes the first 200 (by default) sentences matching a query, scores them according to how good a dictionary example the computer thinks they will make, and returns them in order, best first. The scoring is done with a series of simple rules addressing the considerations listed above: how long is the sentence; does it contain words outside the core language vocabulary; does it begin with a capital letter and end with a full stop, exclamation mark or question mark; does it contain an excessive number of characters other than lower-case a-to-z? The goal is that the average number of corpus lines that a lexicographer has to read, before finding one suitable to use or adapt for the dictionary entry, is substantially reduced, so they rarely have to look beyond the first ten whereas without GDEX, they may often have had to look through thirty or forty.

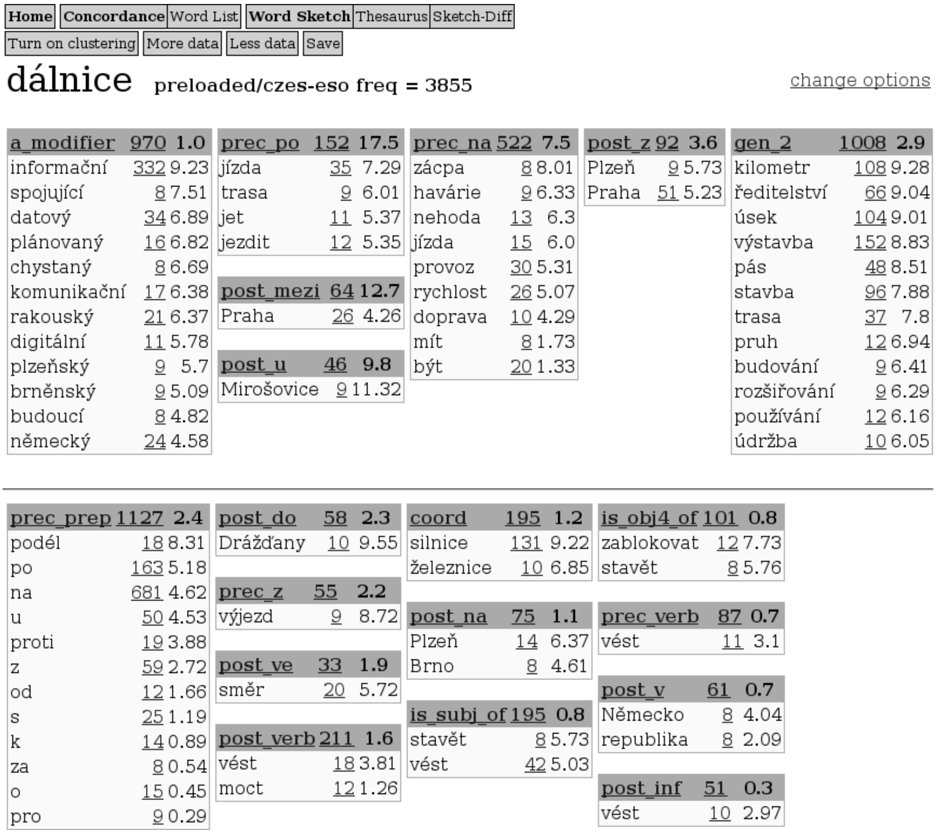
4 Word Sketches and the CZES corpus

Word sketches are the distinctive feature of the Sketch Engine. Word sketches are one-page automatic, corpus-based summaries of a word's grammatical and collocational behaviour. Word sketches improve on standard collocation lists by using a grammar and parser to find collocates in specific grammatical relations, and then producing one list of subjects, one of objects, etc. rather than a single grammatically blind list.

In order to identify a word's grammatical and collocational behaviour, the Sketch Engine needs to know how to find words connected by a grammatical relation. For this to work, the input corpus needs to be parsed or at least POS tagged.

If the corpus is parsed, the information about grammatical relations between words is already embedded in the corpus and the Sketch Engine can use this information directly. A modification of this method was used to handle output of a syntactic parser. If the corpus is POS-tagged but not parsed, grammatical relations can be defined by the developer within the Sketch Engine using a Sketch Grammar.

An example of the word sketch is in Figure 2. The user can set various preferences for the display of the word sketches. Collocates can be ranked according to the frequency of the collocation, or according to its salience score (see [5] for the formula used to compute the salience). The user can set a frequency threshold so low-frequency collocations are not shown. On the results screen the user can go to the related concordance by clicking on the number next to the lemma.



dálnice preloaded/czes-eso freq = 3855

a_modifier	970	1.0
informační	332	9.23
spojující	8	7.51
datový	34	6.89
plánovaný	16	6.82
chystaný	8	6.69
komunikační	17	6.38
rakouský	21	6.37
digitální	11	5.78
plzeňský	9	5.7
brněnský	9	5.09
budoucí	8	4.82
německý	24	4.58

prec_po	152	17.5
jízda	35	7.29
trasa	9	6.01
jet	11	5.37
jezdit	12	5.35

post_mezi	64	12.7
Praha	26	4.26

post_u	46	9.8
Mirošovice	9	11.32

prec_na	522	7.5
zácpa	8	8.01
havárie	9	6.33
nehoda	13	6.3
jízda	15	6.0
provoz	30	5.31
rychlost	26	5.07
doprava	10	4.29
mít	8	1.73
být	20	1.33

post_z	92	3.6
Plzeň	9	5.73
Praha	51	5.23

gen_2	1008	2.9
kilometr	108	9.28
ředitelství	66	9.04
úsek	104	9.01
výstavba	152	8.83
pás	48	8.51
stavba	96	7.88
trasa	37	7.8
pruh	12	6.94
budování	9	6.41
rozšiřování	9	6.29
používání	12	6.16
údržba	10	6.05

prec_prep	1127	2.4
podél	18	8.31
po	163	5.18
na	681	4.62
u	50	4.53
proti	19	3.88
z	59	2.72
od	12	1.66
s	25	1.19
k	14	0.89
za	8	0.54
o	15	0.45
pro	9	0.29

post_do	58	2.3
Drážďany	10	9.55

prec_z	55	2.2
výjezd	9	8.72

post_ve	33	1.9
směr	20	5.72

post_verb	211	1.6
vést	18	3.81
moct	12	1.26

coord	195	1.2
silnice	131	9.22
železnice	10	6.85

post_na	75	1.1
Plzeň	14	6.37
Brno	8	4.61

is_subj_of	195	0.8
stavět	8	5.73
vést	42	5.03

is_obj4_of	101	0.8
zablokovat	12	7.73
stavět	8	5.76

prec_verb	87	0.7
vést	11	3.1

post_v	61	0.7
Německo	8	4.04
republika	8	2.09

post_inf	51	0.3
vést	10	2.97

Fig. 2. Word sketch for the word “dálnice” (highway).

4.1 Czech Sketch Grammar

In this model, grammatical relations are defined as regular expressions over POS-tags. For example, a grammatical relation specifying the relation between a noun and a pre-modifying adjective looks like this.

```
=modifier
2:"A.*" 1:"N.*"
```

The first line, following the =, gives the name of this grammatical relation. The `1:` and `2:` mark the words to be extracted as first argument (the keyword) and second argument (the collocate).

The result is a regular expression grammar which we call a Sketch Grammar. It allows the system to automatically identify possible relations of words to the keyword. These grammars are of course less than perfect, but given the errors in the POS-tagging, this is inevitable however good the grammar. The problem of noise is mitigated by the statistical filtering which is central to the preparation of word sketches.

The first version of the Czech Sketch Grammar was created in the early stage of the Sketch Engine development [1]. It was prepared for the "Prague" tag-set used in the Czech National Corpus. We have adopted the grammar to match the Brno annotation.

When the corpus is parsed with the grammar, the output is a set of tuples, one for each case where each pattern matched. The tuples comprise (for the two-argument case), the grammatical relation, the headword, and the collocate, as in the third column in the table. This work is all done on lemmas, not word forms, so headword and collocate are lemmas.

As can be seen from Table 1, grammatical relations in the Czech Sketch Grammar are of four types, i.e. regular (one way dependency relation), symmetric (between two items with equal status), dual (between two items with dependent relations), trinary (between three dependent items). The sketch engine also supports unary relations but these are not used in the Czech Sketch Grammar. Unary relations are used to extract certain complementation patterns. For instance, a lexicographer would like to know that a verb is frequently followed by a relative clause starting with *že* (that) or that a noun is preceded by an article or not.

Dual relations are the most common. They work similarly to symmetric relations but inversing a dual relation results in a different grammatical relation. A typical dual is the pair, "verb and its object" and "noun and the verb it is object of".

Figure 2 shows the resulting word sketch for word *dálnice* (highway).[4] We can see that the discovered collocations can say a lot about the document sources – here, the most frequent adjective modifier of *dálnice* is *informační* (information highway). An interesting evidence of the state of Czech highways is the list corresponding to the preposition *na* (at), which contains *zácpa* (traffic jam), *havárie* (crash) and *nehoda* (accident) as its top entries.

The Czech Sketch Grammar generates about 46 million triples (dependences) from the 85 million token corpus.

[4] The word sketch is about two times bigger with the default options.

Table 1. The Czech Sketch Grammar grammatical relations

Relation	Example	Triple(s)
symmetric relations		
COORD	silnice a dálnice *roads and highways*	⟨coord,silnice,dálnice⟩ ⟨coord,dálnice,silnice⟩
regular relations		
PREC_VERB	v blízkosti vede dálnice *a highway is nearby*	⟨prec_verb,dálnice,vést⟩
POST_VERB	dálnice většinou vede obcemi *highway usually goes through cities*	⟨post_verb,dálnice,vést⟩
POST_INF	kudy měla nová dálnice vést *where should the new highway go*	⟨post_inf,dálnice,vést⟩
PREC_PREP	telefony podél dálnic *phones along highways*	⟨prec_prep,dálnice,podél⟩
POST_PREP	dálnice před Prahou *the highways in front of Prague*	⟨post_prep,dálnice,před⟩
dual relations		
IS_SUBJ_OF/HAS_SUBJ	kudy dálnice povede *where will the highway go*	⟨is_subj_of,dálnice,vést⟩ ⟨has_subj,vést,dálnice⟩
IS_OBJ2_OF/HAS_OBJ2	co se týká dálnice *what applies to highway*	⟨is_obj2_of,dálnice,týkat se⟩ ⟨has_obj2,týkat se,dálnice⟩
IS_OBJ3_OF/HAS_OBJ3	situace přinese dálnici ... *the situation brings new possibilities to the highway*	⟨is_obj3_of,dálnice,přinést⟩ ⟨has_obj3,přinést,dálnice⟩
IS_OBJ4_OF/HAS_OBJ4	kamion zablokoval dálnici *truck blocked the highway*	⟨is_obj4_of,dálnice,zablokovat⟩ ⟨has_obj4,zablokovat,dálnice⟩
IS_OBJ7_OF/HAS_OBJ7	vládá se zabývá dálnicemi *the government deals with highways*	⟨is_obj7_of,dálnice,zabývat se⟩ ⟨has_obj7,zabývat se,dálnice⟩
GEN_1/GEN_2	dálnice budoucnosti *highway of the future*	⟨gen_1,dálnice,budoucnost⟩ ⟨gen_2,budoucnost,dálnice⟩
PASSIVE/SUBJ_OF_PASSIVE	přeplněná dálnice *crowded highway*	⟨passive,přeplnit,dálnice⟩ ⟨subj_of_passive,dálnice,přeplnit⟩
CATEG1/CATEG2	dálnice je typ silnice *highway is a type of a road*	⟨categ1,dálnice,silnice⟩ ⟨categ2,silnice,dálnice⟩
AJINE1/AJINE2	dálnice a jiné projekty *highways and other projects*	⟨ajine1,dálnice,projekt⟩ ⟨ajine2,projekt,dálnice⟩
BYT_ADJ/SUBJ_BYT	dálnice byla namrzlá *the highway was frosty*	⟨byt_adj,namrzlý,dálnice⟩ ⟨subj_byt,dálnice,namrzlý⟩
A_MODIFIER/MODIFIES	informační dálnici *information highway*	⟨a_modifier,dálnice,informační⟩ ⟨modifies,informační,dálnice⟩
trinary relations		
POST_*	na dálnici v Německu, *at the highway in Germany*	⟨post_*,dálnice,Německo,v⟩ ⟨post_v,dálnice,Německo⟩
PREC_*	u výjezdu z dálnice *at the highway exit*	⟨prec_*,dálnice,výjezd,z⟩ ⟨prec_z,dálnice,výjezd⟩

4.2 Dependency Relations from Syntactic Parser

The Czech syntactic parser `synt` [6,7] is developed in the Natural Language Processing Centre at Masaryk University. The parsing system uses an efficient variant of the head driven chart parsing algorithm [8] together with the meta-grammar formalism for the language model specification. The advantage of the meta-grammar concept is that the grammar is transparent and easily maintainable by human linguistic experts. The meta-grammar includes about 200 rules covering both the context-free part as well as context relations. Contextual phenomena (such as case-number-gender agreement) are covered using the per-rule defined contextual actions. The meta-grammar serves as a basis for a machine-parsable grammar format used by the actual parsing algorithm – this grammar form contains almost 4,000 rules.

Currently, the `synt` system offers a coverage of more than 92 percent of (common) Czech sentences[5] while keeping the analysis time on the average of 0.07s/sentence.

Besides the standard results of the chart parsing algorithm, `synt` offers additional functions such as partial analysis (shallow parsing) [10], effective selection of *n*-best output trees [8], chart and trees linguistic simplification [11], or extraction of syntactic structures [12]. All these functions use the internal chart structure which allows to process potentially exponential number of standard derivation trees still in polynomial time.

Apart from the common generative constructs, the metagrammar includes feature tagging actions that specify certain local aspects of the denoted (non-)terminal. One of these actions is the specification of the head-dependent relations in the rule — the `depends()` construct:

```
/* černá kočka (black cat) */
np → left_modif np
    depends($2,$1)
/* třeba (perhaps) */
part → PART
    depends(root,$1)
```

In the first rule, `depends($2,$1)` says that (the head of) the group under the `left_modif` non-terminal depends on (the head of) the `np` group on the right hand side. In the second example, `depends(root,$1)` links the `PART` terminal to the root of the resulting dependency tree. The meta-grammar allows to assign *labels* to parts of derivation tree, which can be used to specify dependencies "crossing" the phrasal boundaries. The `synt` system thus allows to process even *non-projective phenomena*, which would otherwise be problematic within a purely phrasal approach.

The relational `depends` actions sequentially build a graph of dependency links between surface tokens. Each call of the action adds a new edge to the graph with the following information about the *dependent* group:

1. the non-terminal at the top of the group (`left_modif` or `np` in the example above),
2. the pre-terminal (word/token category) of the *head* of the group, i.e. the single token representing the group, and
3. the grammatical case of the head/group, if applicable.

[5] measured on 10,000 sentences from the DESAM corpus [9].

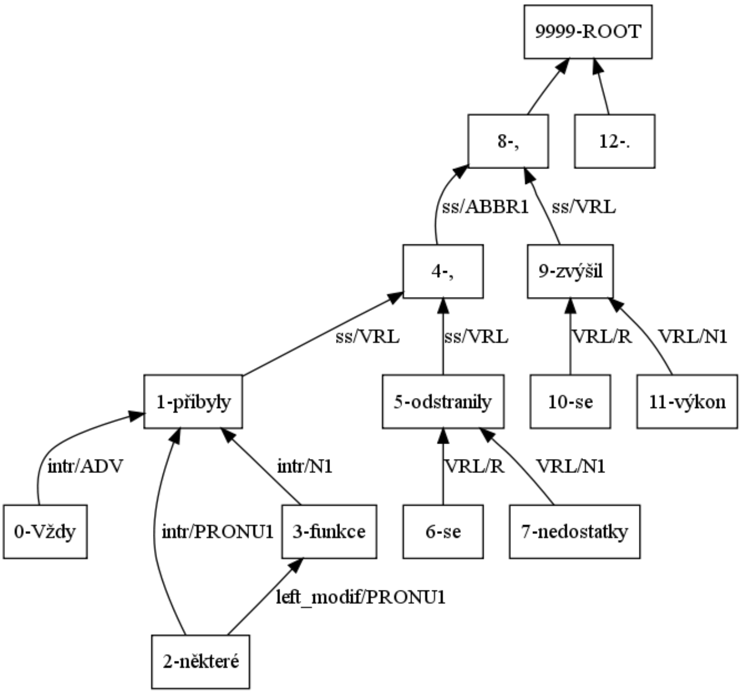


Fig. 3. An example of `synt` dependency graph output for the sentence "*Vždy přibyly některé funkce, odstranily se nedostatky, zvýšil se výkon.*" (Each time new functions were added, drawbacks were removed, the power increased).

An example list of such dependency relations for a corpus sentence "*Vždy přibyly některé funkce, odstranily se nedostatky, zvýšil se výkon.*" (Each time new functions were added, drawbacks were removed, the power increased) may look like this:

from	*label*	*to*	*from*	*label*	*to*
0	`intr/ADV`	1	5	`ss/VRL`	4
1	`ss/VRL`	4	6	`VRL/R`	5
2	`intr/PRONU1`	1	7	`VRL/N1`	5
2	`left_modif/PRONU1`	3	9	`ss/VRL`	8
3	`intr/N1`	1	10	`VRL/R`	9
4	`ss/ABBR1`	8	11	`VRL/N1`	9

The corresponding dependency graph of this sentence is depicted in Figure 3.

We can see that the information in these relations contains more details that come from the parsing process. However, not all details bring the same amount of linguistic adequacy – e.g. distinguishing `left_modif/ADJ1` and `left_modif/ADJ2` does not bring any new information,[6] whereas `intr/N1` links to verbs where the dependent group is a subject and `intr/N4` lists objects in accusative.

Within the experiment of parsing the CZES corpus (about 4 million sentences), we have obtained more than 52 millions of dependency relations.

4.3 Thesaurus

Once the corpus has been parsed and the tuples extracted, we have a very rich database that can be used in a variety of ways.

We can ask "which words share most tuples", in the sense that, if the database includes both $\langle gramrel, w_1, w\rangle$ and $\langle gramrel, w_2, w\rangle$ (for example $\langle \text{prec\_na}, \text{dálnice}, \text{provoz}\rangle$ and $\langle \text{prec\_na}, \text{silnice}, \text{provoz}\rangle$), then we can say that w_1 and w_2 share a triple. A shared triple is a small piece of evidence that two words are similar. Now, if we go through the whole lexicon, asking, for each pair of words, how many triples do they share, we can build a 'distributional thesauruses', which, for each word, lists the words most similar to it (in an approach pioneered in [13,14]). The Sketch Engine computes such a thesaurus. A thesaurus entry for *dálnice* obtained from the standard Sketch Grammar starts with:[7]

- silnice (road)
- železnice (railway)
- trasa (path), trat' (route), most (bridge)
- elektrárna (power station), komunikace (communication), vozovka (pavement), ropovod (pipeline)
- infrastruktura (infrastructure)

The same thesaurus entry computed with the dependency relations obtained from syntactic parsing looks like:

- silnice (road)
- ropovod (pipeline), tunel (tunnel), trasa (path)
- vozovka (pavement), infrastruktura (infrastructure), most (bridge), železnice (railway), trat' (route), komunikace (communication)
- dráha (line)
- elektrárna (power station)

The main synonym *silnice* stays the same, but other similar words are grouped in different order. Evaluation of these two approaches, however, needs further studies from both grammarian and lexicographer's point of view.

5 Conclusion

We have loaded the CZES corpus into the Sketch Engine. The process was designed to support various lexicographic tasks at the Masaryk University NLP Centre.

The distinctive feature of the Sketch Engine are its word sketches. The standard way to set them up for Czech involved writing a Sketch Grammar to define the set of Czech

[6] It just says that the collocation *adjective+noun* was in nominative or genitive.

[7] The words are grouped according to the thesaurus score.

Grammatical relations. Each grammatical relation is defined using a regular-expression grammar over part-of-speech tags. The paper documents the grammatical relations for Czech.

Another way of defining word sketches, that was experimentally tested, lies in using dependency relations obtained from full syntax parsing of Czech. The resulting dependency relations provide further levels of details coming from the parsing process at the place of the relation label. What remains to be done is to prepare a linguistically justified translation of these labels to provide the most adequate information based on the parsing results.

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 407/07/0679.

References

1. Kilgarriff, A., Rychlý, P., Smrž, P., Tugwell, D.: The Sketch Engine. Proceedings of Euralex (2004) 105–116. `http://www.sketchengine.co.uk`
2. Šmerk, P.: Towards czech morphological guesser. In: Proceedings of Recent Advances in Slavonic Natural Language Processing 2008, Brno, Czech Republic, Masaryk University (2008) 1–4
3. Sedláček, R.: Morphemic Analyser for Czech. PhD thesis, Masaryk University (2005)
4. Kilgarriff, A., Husák, M., McAdam, K., Rundell, M., Rychlý, P.: GDEX: Automatically Finding Good Dictionary Examples in a Corpus. In: Proceedings of the XIIIth EURALEX International Congress. Barcelona: Universitat Pompeu Fabra. (2008) 425–432
5. Rychlý, P.: A Lexicographer-Friendly Association Score. In: Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2008, Brno, Czech Republic, Masaryk University (2008) 6–9
6. Horák, A.: The Normal Translation Algorithm in Transparent Intensional Logic for Czech. PhD thesis, Masaryk University (2002)
7. Kadlec, V., Horák, A.: New Meta-grammar Constructs in Czech Language Parser synt. In: Lecture Notes in Computer Science, Springer Berlin / Heidelberg (2005)
8. Horák, A., Kadlec, V., Smrž, P.: Enhancing Best Analysis Selection and Parser Comparison. In: Lecture Notes in Artificial Intelligence, Proceedings of TSD 2002, Brno, Czech Republic, Springer Verlag (2002) 461–467
9. Pala, K., Rychlý, P., Smrž, P.: DESAM – Annotated Corpus for Czech. In: Proceedings of SOFSEM '97, Springer-Verlag (1997) 523–530
10. Ailomaa, M., Kadlec, V., Rajman, M., Chappelier, J.C.: Robust stochastic parsing: Comparing and combining two approaches for processing extra-grammatical sentences. In Werner, S., ed.: Proceedings of the 15th NODALIDA Conference, Joensuu 2005, Joensuu, Ling@JoY (2005) 1–7
11. Kovář, V., Horák, A.: Reducing the Number of Resulting Parsing Trees for the Czech Language Using the Beautified Chart Method. In: Proceedings of the 3rd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, Poznan, Poland (2007) 433–437

12. Jakubíček, M., Horák, A., Kovář, V.: Mining Phrases from Syntactic Analysis. In: Proceedings of TSD 2009, Springer-Verlag (2009)
13. Grefenstette, G.: Explorations in automatic thesaurus discovery. Springer (1994)
14. Lin, D.: Automatic Retrieval and Clustering of Similar Words. In: Conference on Computational Linguistics (COLING-ACL). (1998) 768–774