IJCLA VOL. 1, NO. 1-2, JAN-DEC 2010, PP. 185-199
RECEIVED 23/11/09 ACCEPTED 16/01/10 FINAL 10/03/10

Exploiting Charts in the MT Between Related Languages

PETR HOMOLA, VLADISLAV KUBOŇ

Institute of Formal and Applied Linguistics, Czech Rep.

ABSTRACT

The paper describes in detail the exploitation of chart-based methods and data structures in a simple system for the machine translation between related languages. The multigraphs used for the representation of ambiguous partial results in various stages of the processing and a shallow syntactic chart parser enable a modification of a simplistic and straightforward architecture developed ten years ago for MT experiments between Slavic languages in the system Česílko. The number of translation variants provided by the system inspired an addition of a stochastic ranker whose aim is to select the best translations according to a target language model.

1 INTRODUCTION

Using graphs has a long tradition in the field of machine translation (MT). It is very difficult to trace back the first attempts to represent some linguistic phenomena by means of charts, but it is not difficult to find a clear historical example of usefulness of such representation. This example is probably the most famous MT system of all times, the first really successful and commercially exploited system, METEO [1,2]. There were many reasons why METEO worked so well that it served for decades as a positive example for the whole MT community demonstrating that machine translation is possible after all. The formalism used in the system, Colmerauer's Q-systems [3], is definitely among those reasons.

Q-systems are in fact a mechanism for transformation of trees which label the edges of an oriented chart. The transformations are controlled by a grammar which contains declarative rewriting rules. Each rule may be applied to a continuous set of edges and the result of its application is a new edge or a continuous set of new edges starting and ending in the same nodes as the original sequence. The grammar may be divided into more parts which constitute a sequence in which the output of a previous phase (in the form of a chart) serves as an input of the subsequent one. At the end of each phase the system deletes all edges which were used on a left hand side of some rule and the edges which do not constitute a part of a path leading from the starting to the final node. This mechanism thus very naturally cleans all partial results and at the same time it allows to maintain ambiguity whenever it is necessary in between two particular phases.

2 Charts in the MT between Related Languages

Apart from METEO, Q-systems were used as well in one of the first systems of MT between related languages, in the Czech-to-Russian MT system RUSLAN [4]. The ability of the chart-based analyzers to deal with ambiguities at various levels (morphology, syntax, semantics) and to preserve them across the levels (certain morphological ambiguities cannot be resolved without syntactic clues) was fully exploited in this MT system.The system used a traditional transfer-based architecture with full-fledged syntactic analysis involving even some semantics. The relatedness of both languages was not reflected in the architecture of the system.

The last decade witnessed a growing interest in MT between related languages for different language groups—Slavic [5,6], Scandinavian [7], Turkic [8], and languages of Spain [9]. The main advantage of translating between related languages is the possibility to use much simpler means, in most cases some kind of “shallow” methods, most prominently in parsing or in transfer. This is actually the case of experiments for Slavic languages and the languages of Spain, where both systems follow a very simple architecture originally designed for the Czech-to-Slovak system Česílko. A morphological tagger disambiguates the input, individual lemmas and tags are translated and transfered into a target language and a morphological synthesis creates a target language sentence. This rather simplistic approach chosen both in the system Česílko and Apertium has a substantial drawback in the fact that the morphological and lexical ambiguity is solved early in the translation process with all the consequences—the taggers used are still not sufficiently precise (the best taggers for highly inflected languages quite naturally still have precision inferior to their counterparts for English) and thus they introduce translation errors which cannot be removed in the subsequent stages of the translation process. The architecture also does not allow to cope with lexical ambiguity, another source of frequent translation errors.

In the following sections we would like to describe in detail how the exploitation of chart-based methods may improve the MT between closely related languages. The description will concern all important processing stages of the system: morphological analysis, shallow syntactic analysis and transfer. The experiments are conducted on a group of Slavic languages with Czech as a source language and Slovak as a primary target language.

3 Charts in Morphology

As mentioned above, the simplistic architecture of Česílko exploits a morphological tagger for a (complete) disambiguation of ambiguous word-forms. In our experiments we have decided to replace the tagger by a shallow syntactic chart parser which helps to (partially) disambiguate the ambiguous input on the basis of the local context and, at the same time, it preserves those ambiguous variants which cannot be resolved in such a way. In order to keep the ambiguities wherever necessary, our system uses a multigraph (i.e., a graph allowing parallel edges between a pair of nodes).

In morphology, the advantage of the multigraph is obvious especially for highly inflected languages. Individual word-forms are very often ambiguous with regard to the gender, number and case and the possibility to keep all the variants as long as necessary (until the ambiguity is resolved in later stages of the processing), is really an important advantage.

The use of a multigraph in a chart parser also has certain hidden drawbacks which have to be handled by workarounds or tricks. Let us discuss the most crucial issue.

Let us consider the Czech sentence *Starý hrad se tyčí nad řekou* “The old castle towers over the river”. The phrase *starý hrad* is morphologically ambiguous (both forms can be used in both nominative and accusative case). After this phrase has been recognized as the subject of the main verb, we know that the case is nominative in this context. And since there is no other reading where it would be accusative, the parser can remove this wrong reading.

But what would have happened if we had the isolated phrase *staré hrady* “old castles”? There would be again two possible readings (nominative and accusative) which cannot be resolved due to the lack of context. Nevertheless there are still other meanings for each of the words independently (disregarding the dependence between them). In this case, these edges will not be removed during the final cleaning of edges although the parser has analyzed the whole phrase. We can use a simple workaround in this case: we can insert a new edge (*shackle*) between edges which represent two word forms of the input sentence. These artificial edges will link both clusters of edges representing different morphological readings. If there is at least one analysis which connects both words, the parser will remove the shackle during the cleaning phrase and thus only the complete parse will be preserved for further processing because the ‘false’ edges will not lie on a valid path any more and will be deleted as well (the adjective would have more morphological meanings; for the sake of simplicity, the multigraph contains only one edge with different gender).

It is obvious that if we modify the multigraph by adding ‘shackles’ between all edges labelled with morphological information about individual input words we also have to modify all grammar rules accordingly.

4 Charts in Syntax

In this section we would like to discuss typical issues of exploiting the chart parser in a syntactic analysis. One of the most important issues which may substantially reduce the parsing efficiency of chart parsers is their natural tendency to create redundant identical results.

4.1 *Elimination of identical results*

The application of grammar rules to the multigraph is non-deterministic, the rules are being applied in an order which may look very close to random. As a result, the application of several different sequences of rules may lead to identical results, as illustrated in Figure 1:

There are two possible parses:

1. The rule identifying direct objects is applied first, the rule identifying subjects is applied afterwards.
2. The rule identifying subjects is applied first, the rule identifying direct objects is applied afterwards.

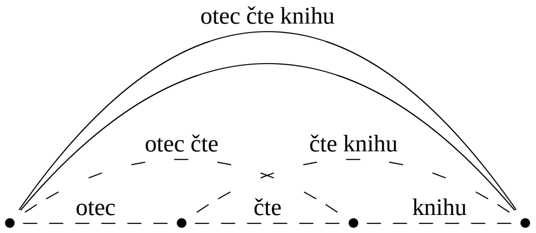

Fig. 1. Example of duplicate parses of a sentence

Theoretically, we would get two edges spanning the whole sentence and labelled with identical dependency trees (of course, if we adhere to constituent trees, both structures will reflect the order of application of grammar rules and they will be different, but let us not forget that we are primarily talking about the MT system between Slavic languages where the use of dependency notation has a long tradition). In our implementation of the parser, this kind of duplicity is recognized automatically to avoid exponential explosion.

4.2 *Multigraph clean-up and further optimization*

As long as a rule can be applied to the multigraph, edges are added to it but no existing edge is removed. The new edges represent (are labelled with) intermediary feature structures that may be used in further parsing or they may be candidates for the final result. Once the multigraph cannot be extended by any rule (according to a particular grammar), the intermediary edges need to be discarded from the multigraph since we want only the most complex feature structures to be processed in the transfer phase. This clean-up is somewhat similar to garbage collection in programming languages with automatic memory management.

As an example, let us consider the following Czech verb phrase as the input of the parser:

(1) *auta* *jezdila*
cars-NEUT,NOM,PL move-PAST,NEUT,PL
"The cars moved/were moving."

The input of the parser is the morphologically preprocessed multigraph (the multisets of edges between the same pair of nodes reflect the morphological ambiguity of a word form), which can be found in the upper part of Figure 2.

After the application of one particular rule, namely the one that attaches a noun in nominative (the subject) to its predicate (a resultative participle in this case), we will get the multigraph from the lower part of Figure 2 as the result of the syntactic analysis (dotted lines denote used edges, circles denote used nodes[1]).

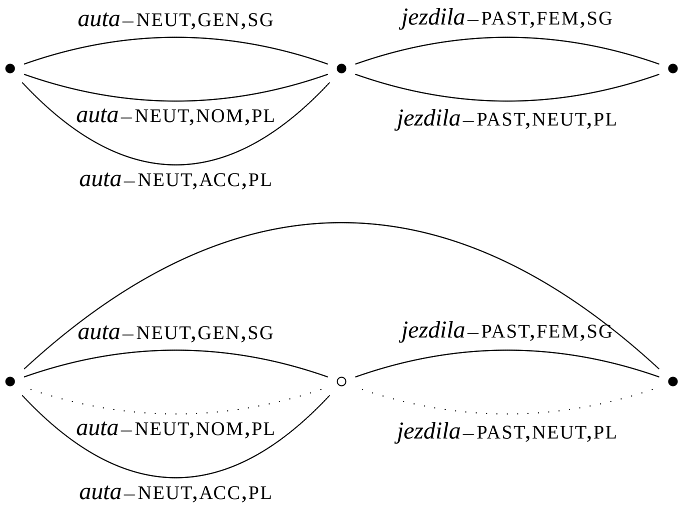

Fig. 2. The input and the result of the syntactic analysis

Now we need to get rid of all obsolete edges:

1. First of all, we remove all used edges (denoted by dotted lines).

[1] We define used node as a node that has at least one used edge to the left and at least one used edge to the right.

2. We remove all edges which start or end in a used node (i.e., the edges that reflect morphological variants of a used edge which are morphologically misanalyzed in the given context according to the used grammar).
3. For each path p from the initial node to the end node, we calculate the number $u(p)$ of used edges it contains. Then we assign each edge e the score $s(e) = \min_{p \in P} u(p)$. The score for the whole graph is defined as $s = \min_{e \in E} s(e)$. Finally, we remove all edges where $s(e) > s$.[2]

The last step ensures that every edge which remains in the multigraph lies on a path from the initial node to the end node. The resulting graph represents the output of the module of shallow syntactic analysis and as such it is passed to the subsequent module which is the transfer. At the same time, all complex feature structures (that represent syntactic trees) that label the edges of the multigraph are being syntactically synthesized.

Processing of long sentences may result in very large multigraphs with the number of edges growing exponentially. If we had to translate the Russian phrase *старый замок* “old castle” into Czech, the transfer would give the two features structures from Figure 3.

$$\begin{bmatrix} \text{”замок”} & \\ \text{ADJ} & [\ \text{”старый”}\ ] \end{bmatrix} \rightarrow \left\{ \begin{bmatrix} \text{”hrad”} & \\ \text{ADJ} & [\ \text{”starý”}\ ] \end{bmatrix}, \begin{bmatrix} \text{”zámek”} & \\ \text{ADJ} & [\ \text{”starý”}\ ] \end{bmatrix} \right\}$$

Fig. 3. Lexical transfer of feature structures

The syntactically synthesized multigraph is shown in Figure 4.

As the two edges with the feature structure for the adjective *starý* are identical, we can optimize the spatial complexity of the multigraph by contracting identical edges that have at least one common node. We call this process *compacting* the multigraph. It is obvious that in complex multigraphs, the number of edges can be lowered significantly. Immediately before morphological synthesis, the optimization can be even more efficient if we do not contract only edges with identical feature structures

[2] If there is at least one path from the initial node to the end node consisting only from unused edges then the algorithm is equal to the one described in [3], i.e., all used edges are deleted as well as edges that do not belong to a path from the initial node to the end node.

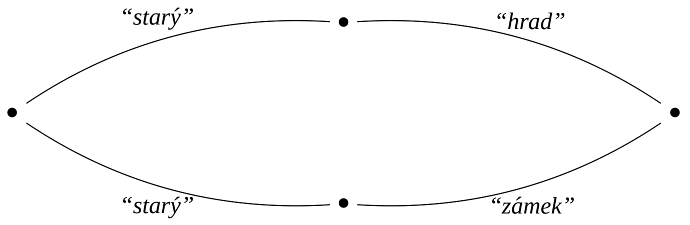

Fig. 4. The result of a transfer and corresponding feature structures

but also those with identical surface form in the target language (there is an extensive syncretism in Slavic languages).

5 TRANSFER AND SYNTACTIC SYNTHESIS

Transfer and syntactic synthesis are performed jointly in one module. The task of the transfer module is to adapt complex structures created by the parser which cover the whole source sentence continuously to the target language lexically, morphologically and syntactically. In the following sections we describe the phase of the lexical transfer and the structural transfer, the latter being split further in structural preprocessor and syntactic decomposer.

5.1 *Lexical transfer*

The aim of the lexical transfer is to 'translate a feature structure lexically', i.e., the lemmas associated with feature structures are translated. Morphological features may be adapted as well wherever appropriate.

In order to demonstrate the nature of the data contained in the dictionary, let us present a fragment of the dictionary used in lexical transfer between Czech and Slovenian:

Example 1.
```
hvězda|zvezda
dodat|dodati
kůň|konj
strom|drevo|gender=neut;
```

Let us have a brief look at the last line of the example. The Czech noun *strom* “tree” is in masculine gender while the gender of its Slovenian counterpart *drevo* is neuter, that is why there is the additional information *gender=neut* which instructs the transfer module to adapt the feature *gender* of the corresponding feature structure so that it can be correctly synthesized morphologically.

5.2 *Structural transfer*

The task of the structural transfer is to adapt the feature structures of the source language (their properties and mutual relationship) so that the synthesis generates a grammatically well-formed sentence with the meaning of the source sentence. It is necessary to admit that the well-formedness can generally be guaranteed only locally for the part of the sentence the feature structure covers (this is caused by the decision to exploit shallow parsing instead of a full-fledged one).

When changing the structure, the transfer may do one of the following actions:

- to change values of atomic features in the feature structure, to add atomic features with a specific value or to delete some atomic features;
- to add a node to the syntactic tree;
- to remove a node from the syntactic tree.

5.3 *Translation of multiword expressions*

It is a well known fact that some words of a source language are translated as multiword expressions in the target language and vice versa, for example:

Example 2. *babička* “grandmother” (Cze) → *stará mama* (Slv)
zahradní jahoda “garden strawberry” (Cze) → *truskawka* (Pol)

Since these cases require removing or adding of a subordinated feature structure (for the adjective) which is equivalent to removing or adding a node from/to the syntactic tree, such cases are handled by special rules in the structural transfer.

6 Ranker

As shown in the previous sections, the multigraph is a very usefull data structure allowing to keep multiple variants of a translated sentence through all stages of automatic translation. In order to fully exploit this property, it is necessary to add a module which would be able to select the best translation from all variants contained in the multigraph. For this purpose we have decided to apply a stochastic ranker. The advantage of using a ranker instead of a tagger is obvious—apart from inserting errors caused by the imperfection of the tagger into the input sentence the tagger also disambiguates the input too early and makes the translation process too straightforward. If we replace it with the ranker we are able to propagate more translation candidates through the system.

The reason why we are using a stochastic module in a system which relies a lot on hand-written rules is pretty obvious—it would be very complicated (if possible at all) to resolve the degree of ambiguity preserved in the multigraph by hand-written rules. The stochastic post-processor is able to select one particular sentence that suits best the given context.

6.1 *Ranking*

We use a simple language model based on trigrams (trained on word forms without any morphological annotation) which is intended to sort out “wrong” target sentences (these include grammatically ill-formed sentences as well as inappropriate lexical mapping). For example, the language model for Slovak has been trained on a corpus of 18.8 million words which have been randomly chosen from the Slovak Wikipedia[3].

Let us present an example of how this component of the system works. Let us suppose thet there is the following Czech segment (matrix sentence) in the source text:

Example 3.	*Společnost*	*ve*	*zprávě*
	company-FEM,SG,NOM	in	report-FEM,SG,LOC

uvedla
inform-LPART,FEM,SG

“The company informed in the report. . . ”

The rule-based part of the system is supposed to generate (the shallow grammar contains no rules for VPs) four Slovak (target) segments that collapse to the following two after morphological synthesis:

[3] http://sk.wikipedia.org

1. *Spoločnosť vo správe uviedli*,
2. *Spoločnosť vo správe uviedla*.

The Czech word *uvedla* is ambiguous (fem.sg and neu.pl). According to the language model, the ranker will choose the second sentence as the most probable result.

There are also many homonymic word forms that result in different lemmas in the target languages. For example, the Czech word *pak* means both "then" and "fool-pl.gen", the word *tři* means "three" and the imperative of "to scrub", *ženu* means "wife-sg.acc" and "(I'm) hurrying out" etc. The ranker is supposed to sort out the contextually wrong meaning in all these cases if it has not been resolved by the parser.

6.2 *Evaluation*

We have evaluated the system of the Czech-to-Slovak MT on hundreds of sentences mainly from newspapers. The metrics we are using is the Levenshtein edit distance between the automatic translation and a reference translation. The reason why we do not use some more standard evaluation metric such as BLEU [10] is simple—there is no sufficiently large set of good quality testing data which would contain multiple translations of each particular source sentence into Slovak. As it has already been shown in several articles (e.g. [11]), the correlation of BLEU with the human judgment is not as high as it was generally believed. On top of that, the reliability of BLEU decreases significantly if only a single reference translation is used. The edit distance has one more advantage—while the BLEU score does not provide any clue how complicated is the post-editing of the result, the Levenshtein metric is pretty straightforward in this respect and thus it is more suitable for really practical evaluation of the MT output.

There are three basic possibilities of the outcome of translation of a segment.

1. The rule-based part of the system has generated a 'perfect'[4] translation (among other hypotheses) and the ranker has chosen it.
2. The rule-based part of the system has generated a 'perfect' translation but the ranker has chosen another one.
3. All translations generated by the rule-based part of the system need post-processing.

[4] By 'perfect' we mean that the result does not need any human post-processing.

In the first case, the edit distance is zero, resulting in accuracy equal to 1. In the second case, the accuracy is $1 - d$ with d meaning the edit distance between the segment chosen by the ranker and the correct translation divided by the length of the segment. In the third case, the accuracy is calculated as for (2) except that we use the reference translation to obtain the edit distance.

Given the accuracies for all sentences we use the arithmetic average as the translation accuracy of the whole text. The accuracy is negatively influenced by several aspects. If a word is not known to the morphological analyzer, it does not get any morphological information which means that it is practically unusable in the parser. Another possible problem is that a lemma is not found in the dictionary. In such a case, the original source form appears in the translation, which naturally decreases the score. Finally, sometimes the morphological synthesis component is not able to generate the proper word form in the target language (due to partial incompatibility of tagsets for both languages). In such a case, the target language (Slovak) lemma appers in the translation.

The results are summarized in Table 1. The results obtained by our system are compared with the results of an original system for Czech-to-Slovak MT. The numbers clearly support the claim that the change of the architecture enabled by an exploitation of a multigraph in all phases of the translation mentioned in our paper improves the system performance. The improvement can be attributed both to the shallow parser as well as the ranker, one without the other provides worse results.

Table 1. Czech-to-Slovak evaluation

accuracy	original	ranker & chunker	ranker & parser
character based	93.9%	96.3%	96.4%
word based	81.1%	87.8%	88.3%

7 Segmentation

Due to morphological, syntactic and lexical ambiguity, the number of edges in a chart may grow exponentially during processing a sentence. Especially for languages with rich inflection, such as Czech and other Slavic languages, this fact may seriously influence the effectivity of the translation process, thus it would be helpful to optimize the processing of sentences that are too long. Since the method described in this paper is based on shallow NLP and parts of source sentences are processed independently, using smaller translation units rather than whole sentences would speed up the translation process without necessarily lowering translation accuracy. In our experiment, we have exploited the corpus of Czech sentences with manually annotated clause structure [12] to see how the segmentation of compound sentences could help.

The Prague Dependency Treebank[5] [13] is a large and elaborated corpus with rich syntactic annotation of Czech newspaper texts. A part of this corpus was manually annotated with respect to structure of sentences—the concept of segments, easily automatically detectable and linguistically motivated units was adopted [14]. Segments are understood as maximal non-empty sequences of tokens that do not contain any punctuation mark or coordinating conjunction. The sentence annotation captures the level of embedding for individual segments. This concept of linear segments serves as a good basis for the identification of clauses—single clause consists of one or more segments with the same level of embedding; one or more clauses then create(s) a complex sentence.

The definition of segments adopted in the project is based on very strict rules for punctuation in Czech. Generally, the beginning and end of each clause must be indicated by a boundary. This holds for embedded clauses as well. In particular, there are only very few exceptions to a general rule saying that there must be some kind of a boundary between two finite verb forms of meaningful verbs.

In the pilot phase of the project, 3,443 sentences from PDT were annotated with respect to their sentence structure which gives 7,975 segments and 5,003 clauses. While most sentences contain only one or two clauses, the maximal observed number of clauses in a sentence is 11.

An experiment that used segments of the corpus instead of whole sentences as translation units has shown that the translation process was 3–4 times faster (depending on the set of syntactic rules) while the accuracy of the translation did not change. Thus the only remaining problem is to refine the algorithm that automatically segments compound sentences of the source language.

[5] `http://ufal.mff.cuni.cz/pdt2.0/`

8 CONCLUSIONS

The results achieved in the experiments with machine translation between two very closely related languages (Czech and Slovak) described in this paper seem to support the hypothesis that the change of the rather simplistic architecture of the original system Česílko enabled by an exploitation of a multigraph and a shallow chart parser combined with a stochastic ranker of the target language sentences generated by the system resulted in improved translation quality. The use of a chart-based technique in several phases of the translation process is a crucial factor for the improvement.

ACKNOWLEDGMENTS

The presented research has been supported by the grant No. 1ET100300517 of the GAAV ČR and partially supported by the grant No. 405/08/0681 of the GAČR.

REFERENCES

1. Chandioux, J.: METEO, an operational system for the translation of public weather forecasts. In: American Journal of Computational Linguistics, FBIS Seminar on Machine Translation, Rosslyn, Virginia (1976) 27–36
2. Thouin, B.: The METEO system. In: Proceedings of a conference Practical experience of machine translation. Ed.Veronica Lawson (Amsterdam, New York, Oxford: North-Holland Publishing Company, 1982), London, England (1981) 39–44
3. Colmerauer, A.: Les systèmes Q ou un formalisme pour analyser et synthétiser des phrases sur ordinateur. Technical report, Mimeo, Montréal (1969)
4. Oliva, K.: A parser for Czech implemented in Systems Q. Technical report, MFF UK, Prague (1989)
5. Marinov, S.: Structural Similarities in MT: A Bulgarian-Polish case (2003)
6. Homola, P., Kuboň, V.: A translation model for languages of acceding countries. In: EAMT Workshop, Malta (2004)
7. Dyvik, H.: Exploiting Structural Similarities in Machine Translation. Computers and Humanities **28** (1995) 225–245
8. Altintas, K., Cicekli, I.: A Machine Translation System between a Pair of Closely Related Languages. In: Proceedings of the 17th International Symposium on Computer and Information Sciences (ISCIS 2002), Orlando, Florida (2002) 192–196

9. Corbi-Bellot, A., Forcada, M., Prtiz-Rojas, S., Perez/Ortiz, J.A., Remirez-Sanchez, G., Martinez, F.S., Alegria, I., Mayor, A., Sarasola, K.: An Open-Source Shallow-Transfer Machine Translation Engine for the Romance Languages of Spain. In: Proceedings of the 10th Conference of the European Association for Machine Translation, Budapest (2005)
10. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, Pennsylvania (2001) 311–318
11. Callison-Burch, C., Osborne, M., Koehn, P.: Re-evaluating the Role of BLEU in Machine Translation Research. In: Proceedings of the EACL'06, Trento, Italy (2006)
12. Lopatková, M., Klyueva, N., Homola, P.: Annotation of sentence structure; capturing the relationship among clauses in czech sentences. In: Proceedings of the Third Linguistic Annotation Workshop (LAW III), Suntec, Singapore, Association for Computational Linguistics (2009) 74–81
13. Hajič, J., Hajičová, E., Panevová, J., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M.: Prague Dependency Treebank 2.0. LDC (2006)
14. Kuboň, V., Lopatková, M., Plátek, M., Pognan, P.: A Linguistically-Based Segmentation of Complex Sentences. In Wilson, D., Sutcliffe, G., eds.: Proceedings of FLAIRS Conference, AAAI Press (2007) 368–374

PETR HOMOLA
INSTITUTE OF FORMAL AND APPLIED LINGUISTICS,
CZECH REP.
E-MAIL: <HOMOLA@UFAL.MFF.CUNI.CZ>

VLADISLAV KUBOŇ
INSTITUTE OF FORMAL AND APPLIED LINGUISTICS,
CZECH REP.
E-MAIL: <VK@UFAL.MFF.CUNI.CZ>