From Czech Morphology through Partial Parsing to Disambiguation

Eva Mráková and Radek Sedláček

NLP Laboratory, Faculty of Informatics, Masaryk University
Botanická 68, CZ-602 00 Brno, Czech Republic
E-mail: {glum,rsedlac}@fi.muni.cz

Abstract. This paper deals with a complex system of processing raw Czech texts. Several modules were implemented which perform different levels of processing. These modules can easily be incorporated into many other linguistic applications and some of them are already exploited in this way. The first level of processing raw texts represents a reliable morphological analysis – we give a survey of the effective implementation of the robust morphological analyser for Czech named `ajka`. Texts tagged by `ajka` can be further processed by the partial parser Dis and its extension VaDis which is based on verb valencies. The output of these systems serves for automatic partial disambiguation of input texts. The tools described in this paper are widely used for parsing large corpora and can be employed in the initial phase of semantic analysis.

Keywords: morphological analysis, partial syntactic parsing, disambiguation, verb valencies

1 Introduction

Czech belongs to a family of highly inflectional free-word order languages. These characteristics demand special treatment during text processing systems of Czech words and sentences.

In analytical languages a simple approach can be taken in morphological analysis: usually it is enough to list all word forms to capture most morphological processes. In English, for example, a regular verb has usually only 4 distinct forms, and irregular ones have at most 8 forms. On the other hand, highly inflected languages like Czech and Turkish [7] present a difficulty for such simple approaches as the expansion of the dictionary is at least an order of magnitude greater; a Czech verb, for instance, can have up to fifty distinct forms.

Chunking in Czech is also more difficult than in English. There are two reasons for this: first, a gap within a verb group may be more complex and it may even be a whole clause. Second, Czech language is a free word-order language which requires a much more complex approach to recognising the verb group structure.

Statistical methods of disambiguation [3] are suitable for analytical languages like English, but the problem of spare learning data arises for languages like Czech with a huge amount of possible morphological tags (compare approx.

1600 tags in `ajka` to 160 tags in the BNC extended tagset). Thus, it turns out that rule-based methods [5] have to be developed and employed to obtain better results.

2 Morphological Analyser `ajka`

We developed a universal morphological analyser which performs the morphological analysis based on dividing all words in Czech texts into their smallest relevant components that we call *segments*. We define several types of segments – most of them roughly corresponds to the linguistic concept *morpheme* (e.g. ending) and some of them represents the combinations of two or more morphemes (e.g. stem).

Our morphological analyser consists of three major parts: a formal description of morphological processes via morphological patterns; an assignment of Czech stems to their relevant patterns; and a morphological analysis algorithm.

The description of Czech formal morphology is represented by a system of inflectional patterns with sets of endings and it includes the lists of segments and their proper combinations. The assignment of Czech stems to their patterns is contained in the Czech Machine Dictionary [6]. Finally, the algorithm of morphological analysis using this information splits each word into its appropriate segments.

2.1 Description of Czech Morphology

The main part of the algorithmic description of formal morphology, as it was suggested in [6], is a pattern definition. The basic notion is a *morphological paradigm* – a set of all forms of the lemma expressing a system of its respective grammatical categories.

As stated in [2], the traditional grammar of Czech suggests a much smaller paradigm system than exists in reality. For this reason we decided to build a large set of paradigm patterns to cover all the variations of Czech from scratch. Fortunately, we were not limited by technical restrictions, which allowed us to follow a straightforward approach to a linguistically adequate and robust solution.

Noun, adjective, pronoun and numeral decline for case and number. Verbs conjugate for person and number and have paradigms for different tenses (present, past, etc.) For example, the noun *blecha* (flea) displays the following forms in the singular paradigm: *blecha* (Nom.), *blechy* (Gen.), *blee* (Dat.), *blechu* (Acu.), *blecho* (Voc.), *blee* (Loc.), *blechou* (Ins.) and another seven forms for the plural paradigm.

The corresponding word forms within each paradigm have the same ending and that allows us to divide the given word form into two parts: a *stem* and an *ending*. For the word *blecha* we obtain the following segmentation: *blech*-{*a,y,u,o,ou*}, *ble*-{*e,e*}.

We introduced a system of ending sets and distinguish two types:*basic* and *peripheral* ending sets. The basic ones (in our example {*a,y,u,o,ou*}) contain

endings that do not influence the form of the stem, while endings from the peripheral ending sets (e.g. {*e,e*}) cause changes in the stem. These changes occur regularly and represent typical alternations in the last letter (*ch-*) or in the final group of the stem.

Every ending carries values of grammatical categories of the relevant word form. These values are encoded in the form of a grammatical tag which is assigned to the respective ending. Thus, ending sets contain pairs of the ending and the appropriate tag.

In the next step, because of possible alternations, we perform further segmentation of stems into a *stem base* (e.g. *ble*) and an *intersegment* (e.g. *ch,*). The stem base is the part that is common to all word forms in the paradigm, i.e. it doesn't change, and the intersegment is a final group of the stem whose form changes.

A *pattern* definition then stores the information about the only possible combinations of a stem base, intersegments and endings (e.g. *ble-¡ch¿-*{*a,y,u,o,ou*}*, ¡¿-*{*e,e*}). From this point of view, our system of declension and conjugation patterns is considered to be a complete and systematic description of all the alternations that can occur in the inflection process. Moreover, this approach allows us not to store all word forms from the paradigm, but only the common stem base assigned to the relevant pattern.

2.2 Implementation of the Analyser

The key to the successful implementation of the analyser is an efficient storage mechanism for lexical items. A trie structure [4] is used for storing stem bases of Czech word forms. One of the main disadvantages of this approach is high memory requirements. We attempted to solve this problem by implementing the trie structure in the form of a minimal finite state automaton. This incremental method of building such an automaton was presented in [1] and is fast enough for our purpose. Moreover, the memory requirements for storing the minimal automaton are significantly lower (see Table 2).

There are two binary files that are essential for the analyser. One of them contains definitions of sets of endings and morphological patterns; its source is a plain text file. The second is a binary image of the Czech Machine Dictionary [6] and contains stem bases and auxiliary data structures. We developed a program `abin` that can read both of these text files and efficiently store their content into appropriate data structures in destination binary files.

The analyser's first step is loading these binary files. These files are not further processed – they are only loaded into memory. This is mainly to allow as quick a start of the analyser as possible.

The next steps of the analyser are determined by those within the morphological analysis algorithm. The basic principle of the algorithm is based on the segmentation described in Section 2.1. The separated ending then determines values of grammatical categories. More details can be found in [10]. Another feature of the analyser is a possibility to select various forms of the basic word form, the so called *lemma*.

Finally, the user can have more versions of binary files that contain morphological information and stem bases, and can specify which pair should be used by the analyser. Users can take advantage of this feature to "switch on" an analysis of colloquial Czech, domain-specific texts etc.

Table 1 shows the sentence *Já jsem se té přednášky zúčastnila. (I have participated in that lecture.)* fully but ambiguously morphologically analysed by `ajka`. To explain the output, for example, the tag `k5eApFnStMmPaP` of the word

Já	<l>já
	<c>k3xPnSc1p1
	<c>k1gNnSc1 <c>k1gNnSc4
	<c>k1gNnSc5 <c>k1gNnPc2
	<c>k1gNnSc6 <c>k1gNnSc7
	<c>k1gNnSc3 <c>k1gNnSc2
	<c>k1gNnPc6 <c>k1gNnPc3
	<c>k1gNnPc1 <c>k1gNnPc4
	<c>k1gNnPc5 <c>k1gNnPc7
jsem	<l>být
	<c>k5eAp1nStPmIaI
se	<l>sebe
	<c>k3xXnSc4p2 <c>k3xPnSc4p2
	<l>s
	<c>k7c7
té	<l>ten
	<c>k3xDgFnSc2 <c>k3xDgFnSc3
	<c>k3xDgFnSc6
	<l>té
	<c>k1gNnSc1 <c>k1gNnSc4
	<c>k1gNnSc5 <c>k1gNnPc2
	<c>k1gNnSc6 <c>k1gNnSc7
	<c>k1gNnSc3 <c>k1gNnSc2
	<c>k1gNnPc6 <c>k1gNnPc3
	<c>k1gNnPc1 <c>k1gNnPc4
	<c>k1gNnPc5 <c>k1gNnPc7
přednášky	<l>přednáška
	<c>k1gFnSc2 <c>k1gFnPc1
	<c>k1gFnPc4 <c>k1gFnPc5
zúčastnila	<l>zúčastnit
	<c>k5eApNnPtMmPaP
	<c>k5eApFnStMmPaP

Table 1. Example of the sentence analysed by `ajka`.

zúčastnila (*participated*) means: part of speech (`k`) is verb (`5`), negation (`e`) is affirmative (`A`), person (`p`) is feminine (`F`), number (`n`) is singular (`S`), tense (`t`) is past (`M`), modus (`m`) is participium (`P`) and aspect (`a`) is perfective (`p`). Lem-

mata and possible tags are prefixed by `<l>`, `<c>` respectively. This example also depicts both lemma and tag ambiguity of Czech word forms. The first one is for instance represented by the word form `se` which belongs to two possible lemmata – `sebe` (reflexive pronoun) and `s` (preposition); the second one by the word form `zúčastnila` which has two alternative tags for the same lemma – for neuter plural and feminine singular.

The power of the analyser can be evaluated by two features. The most important is the number of words that can be recognised by the analyser. This number depends on the quality and richness of the dictionary. Our database contains 223,600 stem bases from which `ajka` is able to analyse and generate 5,678,122 correct Czech word forms. The second feature is the speed of analysis. In the brief mode, `ajka` can analyse more than 20,000 words per second on PentiumIII processor with a frequency of 800MHz. Some other statistical data, such as number of segments and size of binary files, is shown in the Table 2.

#intersegments	779
#endings	643
#sets of endings	2,806
#patterns	1,570
#stem bases	223,600
#generated word forms	5,678,122
#generated tags	1,604
speed of the analysis	20,000 words/s
dictionary	1,930,529 Bytes
morph. information	147,675 Bytes

Table 2. Statistical data

3 Partial Parser Dis

The partial parser Dis consists of a robust grammar for the main sentence groups in Czech – verb, nominal and prepositional – and the parsing mechanism. As mentioned above, chunking in Czech is quite difficult particularly because of some properties of verb groups.

Thus the main focus is put on the method for obtaining verb rules from an annotated corpus. One of the most important results of our work is a complete and robust algorithmic description of Czech verb groups; an appropriate version of such a description was not elaborated before.

On the other hand, noun and prepositional groups are quite well described in Czech grammars. For the construction of the grammar rules for recognition of such groups we have used existing resources but the rules have been slightly modified according to the corpus data and our requirements (we prefered higher

recall to precision). Further details of the rules for noun and prepositional groups in our system can be found in [11].

3.1 Verb Rules

Recognition and analysis of the predicate in a sentence is fundamental to the meaning of the sentence and its further analysis. In more than 50% of Czech sentences, the predicate contains a compound verb group (e.g. the group *jsem se zúčastnila* in the example presented in the section describing `ajka`). Moreover, compound verb groups in Czech are often split into more parts with so called gap words. In our example the gap words are *té přednášky*. Until all parts of a compound verb group are located, it is impossible to continue with any kind of syntactic or semantic analysis. We consider a compound verb group to be a list of verbs and maybe the reflexive pronouns *se, si*. Such a group is obviously compound of auxiliary and lexical verbs.

We describe here the method that results in definite clause grammar rules – called verb rules – that contain information about all components of a particular verb group and about their respective tags. The algorithm for learning verb rules takes as its input sentences from the annotated and fully disambiguated corpus DESAM [8]. The algorithm is split into three steps: finding verb chunks (i.e. finding boundaries of simple clauses in compound or in complex sentences, and elimination of gap words), generalisation and verb rule synthesis. These three steps are described below.

1. The observed properties of a verb group are the following: their components are either verbs or a reflexive pronoun *se* (*si*); the boundary of a verb group cannot be crossed by the boundary of a sentence; and between two components of the verb group there can be a gap consisting of an arbitrary number of non-verb words or even a whole sentence. In the first step, the boundaries of all sentences are found. Then each gap is replaced by the symbolic tag `gap`. The method exploits only the lemma of each word (nominative singular for nouns, adjectives, pronouns and numerals, infinitive for verbs) and its tag.
2. The lemmata and the tags are now being generalised. Three generalisation operations are employed: elimination of (some of) lemmata, generalisation of grammatical categories and finding grammatical agreement constraints.
   All lemmata except forms of auxiliary verb *být (to be)* (být, by, aby, kdyby) are rejected. Lemmata of modal verbs and verbs with similar behaviour are replaced by the symbolic tag `modal`. These verbs have been found in the list of more than 15 000 verb valencies [9].
   Exploiting linguistic knowledge, several grammatical categories are not important for verb group description (very often it is negation or aspect). These categories may be removed.
   Another situation appears when two or more values of some category are related. In the simplest case they have to be the same, more complicated cases (e.g. the polite way of addressing in Czech) are treated through special predicates.

3. Finally the verb rule in DCG formalism in Prolog is constructed by rewriting the result of the generalisation phase. For the verb group *jsem se té přednášky zúčastnila* which contains the gap *té přednášky* the following rule is constructed:

```
vg(vg(Be,Se,Verb), Gaps) -->
   be(Be,_,P,N,tP,mI,_),
   % jsem
   reflex_pron(Se,xX,_,_),
   % se
   gap([],Gaps),
   % té přednášky
   k5(Verb,_,_,P1,N1,tM,mP,_),
   % zúčastnila
   { check_num(N,N1,Be,Vy) }.
```

The interpretation of non-terminals used in the rule is following: `be()` represents auxiliary verb *být*, `reflex_pron()` stands for reflexive pronoun *se (si)*, `gap()` is a special predicate for manipulation with gaps, and `k5()` stands for arbitrary non-auxiliary verb. The particular values of some arguments of non-terminals represent required properties. Simple cases of grammatical agreement are not present in this example, more complicated situations are solved employing constraints like the predicate `check_num()`. In the comments there are mentioned words processed by the particular non-terminal. The method has been implemented in Perl. More than 150 definite clause grammar rules were constructed from the annotated corpus that describe all the verb groups that are frequent in Czech.

3.2 Parsing Mechanism

Dis exploits standard Prolog DC parsing mechanism extended two ways. First, the special predicate for processing gaps was designed, implemented and incorporated into this mechanism. Second, it was extended by a procedure which controls the whole parsing, calls the DC mechanism when necessary and selects the best parses in the particular context. In our example sentence the partial parser recognises the verb group *jsem se zúčastnila* and the noun group *té přednášky*. It assigns to each word involved in one of these groups the correct tag in the context of the whole sentence. In general, more analyses of one group could be found and on the partial parsing level it is not possible to choose the only correct one. Dis outputs all of these analyses; some of such ambiguities could be solved using its extension called VaDis.

4 Extension VaDis

Techniques of partial parsing exploited by Dis aim to find the syntactic information efficiently and reliably from unrestricted texts by sacrificing completeness and depth of analysis. The main purpose of the extension VaDis is to find more

complex syntactic relations in the output of the partial parser. It is based on the processing of verb valencies and possible functions of nominal and prepositional groups in a sentence.

The list of Czech verb valencies [9] was transformed to the dictionary suitable for the effective processing by VaDis. The full-meaning verb from the verb group found during the preceding partial analysis is searched in this dictionary. There could be several verb frames associated with one verb. A verb frame consists of one or more parts which express the requirements for their possible participants. If there is at least one potential participant for every part of the verb frame in the analysed sentence, this frame is tagged to be possible. Finally we obtain a list of all possible verb frames with all their potential participants. One of these frames is correct in the particular context. Although we are not able to determine the correct frame automatically now, the selection of possible cases is very useful. Combination with other methods can bring even better results in the future. In addition to verb valencies processing, VaDis searches for all possible arguments of other functions of noun and prepositional groups in a sentence. Thus, to every such a group are assigned all its possible functions in the particular sentence.

The following results are obtained for our example input sentence: only one possible verb frame (out of four frames in the dictionary) is selected for the full-meaning verb *zúčastnit se*. The successful frame is `zúčastnit se <čeho>` (participate ¡in st¿) and its only potential participant is the noun group *té přednášky*. The other relation found is that *Já* is the only possible subject of the sentence.

5 Partial Automatic Disambiguation

The output of both Dis and VaDis can serve for automatic partial disambiguation of the input texts. For every word in the analysed sentence the output contains all tags of the particular word which are involved either in a recognised group containing this word or a function which the word can play in the sentence. One of these groups and roles is correct for the word in the given context.

Thus, if we reduce the full set of tags assigned initially to the word by `ajka` to that tags which occur in Dis or VaDis output, the only correct tag remains in the new restricted set. Described method is used for obtaining partially disambiguated texts. In Table 3 is shown our example sentence disambiguated pursuant to the output of VaDis programme.

In this case the sentence is disambiguated fully and every word has assigned its correct tag in the given context. The recall of our system is 99.03% and the precision of 66.10% which are slightly better results in comparison with at present best Czech rule-based disambiguator [5]. Considerably better precision of disambiguation can be achieved by combining our system with efficient statistical component which is unfortunately not readily available for us now.

```
Já          <l>já
            <c>k3xPnSc1p1
jsem        <l>být
            <c>k5eAp1nStPmIaI
se          <l>sebe
            <c>k3xXnSc4p2
té          <l>ten
            <c>k3xDgFnSc2
přednášky   <l>přednáška
            <c>k1gFnSc2
zúčastnila <l>zúčastnit
            <c>k5eApFnStMmPaP
```

Table 3. Disambiguation performed pursuant to VaDis output.

6 Conclusion

The system consisting of modules performing the two particular levels of natural language analysis – morphological and syntactic ones – was described.

The morphological analyser has been tested on large corpora containing approx. 10^8 positions. Based on the test results we consider `ajka` to be robust enough to analyse any raw Czech texts. Nowadays it is being used for lemmatisation and morphological tagging, as well as for generating correct word forms, and also as a spelling checker. Moreover, `ajka` can readily be adapted to other inflectional languages that have to deal with morphological analysis. In general, only the language-specific parts of the system, i.e. definitions of sets of endings and the dictionary, have to be replaced for this purpose.

We have presented a survey of the main features of the partial parser Dis; the main focus has been put on the algorithm for obtaining verb rules for Czech from an annotated corpus. The extension VaDis recognises further syntactic relations in the output of Dis. It is based on the processing of verb valencies and it searches possible arguments of roles of noun and prepositional groups in a sentence.

The results of the partial disambiguation performed by our system are slightly better than results of other comparable rule-based systems for Czech. We reach the recall of 99.03% and precision of 66.10%. A combination of our system with efficient statistical component will probably bring considerably better score of precision.

References

1. Daciuk, J., Watson, R. E. and Watson, B. W. Incremental Construction of Acyclic Finite-State Automata and Transducers. In *Finite State Methods in Natural Language Processing*, Bilkent University, Ankara, Turkey, June – July 1998.
2. Hajič, J. Disambiguation of Rich Inflection (Computational Morphology of Czech). Charles University Press, 1st edition, 2000.

3. Hajič, J. and Hladká, B. Probabilistic and Rule-Based Tagging of an Inflective Language – a Comparsion. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, Washington 1997.
4. Knuth, D. E. *The Art of Computer Programming: Sorting and Searching*, Volume 3, Chapter 6.3. Addison Wesley, 2nd edition, 1973.
5. Oliva, K., Hnátková, M., Petkevič, V. and Květoň, P. The Linguistic Basis of a Rule-Based Tagger of Czech. In *Proceedings of the Third International Workshop TSD 2000*, Springer, Berlin 2000.
6. Osolsobě, K. *Algorithmic Description of Czech Formal Morphology and Czech Machine Dictionary.* Ph.D. Thesis, Faculty of Arts, Masaryk University Brno, 1996. In Czech.
7. Oztaner S. M. A Word Grammar of Turkish with Morphophonemic Rules. Master's Thesis, Middle East Technical University, 1996.
8. Pala, K., Rychlý, P., and Smrž, P. DESAM - Annotated Corpus for Czech. In *Proceedings of SOFSEM'97, LNCS 1338*, Springer, 1997.
9. Pala, K. and Ševeček, P. Valencies of Czech Verbs. Studia Minora Facultatis Philosophicae Universitatis Brunensis, A45, 1997.
10. Sedláček, R. and Smrž, P. Automatic Processing of Czech Inflectional and Derivative Morphology. Technical Report FIMU-RS-2001-03, Faculty of Informatics, Masaryk University Brno, 2001.
11. Žáčková, E. Partial Parsing (of Czech). Ph.D. Thesis, Faculty of Informatics, Masaryk University Brno, 2002. In Czech.