Johanka Doležalová and Vladimír Petkevič, Charles University, Prague
Johanka@ucw.cz
Vladimir.Petkevic@ff.cuni.cz

Shallow Parsing of Czech Sentence Based on Correct Morphological Disambiguation

0. Introduction

This paper describes an approach to shallow parsing of Czech sentence that is based on an almost correct morphological disambiguation. It might be considered an extension of a purely rule-based morphological disambiguation system which makes the disambiguation rules usable for shallow parsing. The paper consists of the following parts:

1. brief description of an automatic morphological disambiguation based on morphosyntactic rules which ensure an (almost) correct morphological disambiguation
2. approach to shallow parsing based on disambiguation rules: implantation of syntactic relations and of delimitation of chunks into the disambiguation rules
3. conclusion summarizing the whole approach and its advantages.

The basic idea of shallow parsing based on a correct disambiguated input consists in dividing the task of shallow parsing into three subtasks:

a) very reliable morphological disambiguation;
b) establishing syntactic links between syntactically related elements, e.g. syntagms formed by a noun and an adjectivcal attribute, linking reflexives linked to their (base) verbs/adjectives/nouns, identification of parts of analytical predicates etc.;
c) chunking.

The basis of such an approach is provided by a very complex and sophisticated rule-based morphological disambiguation which can disambiguate Czech sentence with a very high reliability, i.e. with a minimum number of errors. This is, of course, very important for any language and all the more so for Czech whose ambiguity rate is generally extremely high (as compared e.g. to other Slavic languages). The complexity of disambiguation consists not only in crude part-of-speech (POS) disambiguation but especially in the identification of cases of nouns, adjectives, pronouns and numerals. If disambiguation is (almost) correct then it necessarily prepares a very solid ground for shallow parsing including chunking. We shall show that both disambiguation and shallow parsing can run in parallel.

The approach is demonstrated on the data taken from various corpora of Czech, especially from the corpus SYN2000, the synchronic part of the Czech National Corpus project (cf. Czech National Corpus).

1. A rule-based system of automatic morphological disambiguation of Czech

Czech is probably the most complex Slavic language from the morphological and syntactic point of view. The tagsets reflecting the morphological complexity of Czech are quite extensive: in one of them, out of approximately 4,400 theoretically existing tags more than 2,000 distinct tags are in real use (cf. Hajič 2004). An automatic morphological analysis, no matter how complex it is, can be realized, unlike morphological and POS disambiguation, in an almost error-free way for known words. Contrary to this, the problem of automatic disambiguation is very difficult to cope with and for Czech it has not been solved in a satisfactory way so far; the best stochastic methods report accuracy about 95 % (cf. Hajič 2004 and Votrubec 2005).

The rule-based system of automatic morphological disambiguation of Czech which is currently being developed (cf. Oliva et al. 2000, Petkevič et al. 2002, Petkevič in press) is based on a manual development of negative and positive disambiguation rules that reflect the syntactic system of Czech in a way similar to Karlsson et al. 1995. The negative disambiguation rules consist in discarding all or some of the incorrect POS and morphological interpretation(s) (encoded by tags) of the given word-form, whereas the positive rules select the correct POS and morphological interpretation(s) of the given word-form in a sentence.

The main features of the rule-based system can be summarized as follows:

- rule system captures the system of the given language (de Saussure's *langue*), i.e. Czech, as reflected in *parole*; the method is linguistically based, i.e. it exploits specific features of the language system of Czech. It is based on the cooperation of (morpho)syntactic *disambiguation rules* reflecting the system of language and a *collocation component* responsible for processing various collocations, idioms and various idiosyncrasies of language
- (morpho)syntactic rules are developed on the basis of *linguistic intuition* and *analysis* of textual data contained in corpora of Czech; there is no automatic inferring of a grammar from a corpus
- the rules may be based on unlimited context, but up to now the scope of rules is limited by sentence boundaries
- the rules use both *negative* and *positive facts* about language
- the disambiguation system uses a *reduction method* which consists in the following: the input to the system is the output of the morphological analysis where:
  recall = 100 % (in the fault-free case, i.e. in case the set of lemmas and tags assigned to a given word form contains the correct one)
  precision = lowest possible (maximum number of *incorrect* lemmas and tags is assigned);
  the method tries to retain the maximum *recall* (100 %) simultaneously maximizing *precision* by the following basic operations:
  - the removal of incorrect morphological interpretations down to (in the optimum case) the only correct lemma(s) and tag(s) – this is a *negative approach* (primarily used by the rules)
  - direct identification of the correct(s) tag(s) only – this is a *positive approach* (used both by the rulers and by the collocation component)
- the system needs *no training data* (but it needs relatively well tagged corpora)
- the performance of the system is not deteriorated in case the size of the tagset increases
- the system does not try to "overdisambiguate", i.e. to disambiguate morphologically inherently ambiguous sentences. This means that each corpus position is assigned correct interpretation(s) (i.e. not necessarily the only one, cf. Oliva 2001)
- the rules and the collocation component cooperate as follows: the collocation component comes first; then the rule-based system follows suit and subsequently the collocation component is invoked again and the whole cycle repeats
- the rules are *mutually independent* and *unordered*, and they operate on constantly more and more disambiguated data; each rule is applied until it cannot disambiguate any more, and after all rules have thus been applied the whole bunch of rules (starting

from the first one whichever it is) is applied again till it is detected that in one cycle the data were not changed

- during rule application no overt syntactic structures (such as trees) are built (but it is *possible to establish syntactic links associating a syntagmatic pair and to identify syntactic chunks* as we shall see below)
- *negative n-gram conception* ($n \geq 2$) is often used by the rules, i.e. the system makes use of tuples of incorrect sequences of tags assigned to word forms, i.e. these sequences violate the (mainly syntactic) system of Czech – these negative n-grams can be automatically extracted from already tagged corpora and they can thus be used as an auxiliary means for the development of rules
- the rules' validity can be measured in terms of decades at least because every language is very slow in changing its syntax
- the rules are written in a special programming language *LanGr* (Language for Grammatical Rules, cf. Květoň 2005) which is especially suited for effective and clearly organized development of rules. At present, the system comprises about 2,200 rules, but their number constantly increases. The latest disambiguation results are published in Květoň 2005.

Morphosyntactic disambiguation rules form the core of the whole approach. A disambiguation rule basically consists of four types of components:

- context
- disambiguation area
- report
- disambiguation action.

which are basically related as follows (cf. Květoň 2005; Petkevič et al. 2002):

$cont_1\ disamb_1\ cont_2\ disamb_2 \ldots cont_n\ disamb_n\ cont_{n+1}\ report\ action$
where

$cont_i$	is the description of a *context*
$disamb_i$	is the description of a *disambiguation area* where the actions (see below) are performed (i.e. data are modified)
report	is the report of a disambiguation action performed
action	is a disambiguation action resulting in removing one or more incorrect tags.

The *context* is always unambiguously specified by means of the *IsSafe* specification – a word form or a sequence of word forms in question must have *only* the specified property, i.e. *all of its morphological interpretations* (tags) *must comply with the condition specified*. This means that *context* must always be unambiguously specified and it is *not changed* by the rule application.

The *disambiguation area* is subject to data change and it is specified by means of the *Possible* specification which states that *at least one of the morphological interpretations* (tags) *of the given word form must comply with the condition specified*. The disambiguation area is typically ambiguous and there are in principle two basic actions (operations) which modify the data:

DELETE some (not necessarily all) incorrect interpretation(s) from one or more corpus positions (i.e. word tokens equipped with lemmas and tags)
LEAVE ONLY correct interpretation(s) in one or more corpus positions (i.e. word tokens equipped with lemmas and tags)

In addition to these basic functions there exist in the system also other functions which, in fact, serve as macros for performing more *DELETE* and *LEAVE ONLY* operations at the same time. For instance, one of such key functions is:

UNIFY [CONDITIONALLY] x y IN [gender,number,case]

which has two operands, *x y* (the operands being two corpus positions, each with its own repertory of tags), and *leaves only* those respective values of the gender, number and case attribute in *x* and *y* which are in the intersection of the values of each of the respective attributes for *x* and *y.* The optional argument *CONDITIONALLY* makes unification conditional (i.e. the *UNIFY* operation is performed only if for each respective attribute the intersection is nonempty). This function can be used mainly for the identification of agreement or saturation of valency requirements.

Example 1. A negative bigram

In this example, the exploitation of a negative bigram by the system is presented. The particular negative bigram is specified like this:

Two non-auxiliary finite verbs in the same clause form a negative bigram.

In other words, in any Czech sentence there cannot be two non-auxiliary finite verbs in the same clause (for expository reasons, exceptions are not taken into account here). This observation may be encoded by the following disambiguation rule:

```
/* In Czech, there cannot be two non-auxiliary finite verbs in the same clause */
rule FinVerbs {
// configuration part
ITEM IsSafe ClauseSeparator;
SEQUENCE OF MustNotBe ClauseSeparator;
posfinverb = ITEM Possible FiniteNonAuxVerb;
SEQUENCE OF MustNotBe ClauseSeparator;
safefinverb = ITEM IsSafe FiniteNonAuxVerb;
SEQUENCE OF MustNotBe ClauseSeparator;
ITEM IsSafe ClauseSeparator;
// executive part
DELETE FiniteNonAuxVerb FROM posfinverb;
};
```

The executive part describes actions to be applied to any sentence which meets the conditions of the configuration part. Such a rule can be applied e.g. to the following

Czech sentences:

(1) *Otec*(Noun) *ženu*(Noun | ~~Verb~~) *dobře viděl*(Verb).
E. lit. *Father woman well saw.*
Father saw the woman well.

(2) *Zformulovali*(Verb) *velmi*(Adverb) *rozhodnou*(~~Verb~~ | Adj) *odpověď*(Noun).
E. They formulated a very decisive response.

The rule *FinVerbs* is a typical representative of the negative approach to the language system of Czech. For shallow parsing we shall use primarily positive rules which directly identify (LEAVE ONLY) correct part-of-speech and morphological interpretations of words in input sentences rather than discard (DELETE) incorrect ones.

2. Shallow Parsing of Czech based on disambiguation

Czech is a prominent representative of inflective Slavic languages. For both historical, theoretical and practical reasons most of the parsers used for Czech are adapted to produce structures based on *dependency syntax*. Up to now, the best results are reported by statistical parsers, in particular, by the Collins's parser (see Collins et al. 1999) and Charniak's parser (the English version is described in Charniak 2000, its application to Czech is not published).

The evaluation of dependency parsers of Czech is simple because standard testing data is available (cf. Hajič et al. 2001a) and the dependency structures are more easily mutually comparable than syntactic structures formed by immediate constituents. A dependency parser has to assign exactly one governing node (parent word) to each node, so there is no need for precision and recall and a single metric called accuracy is used.

Accuracy of the best statistical parsers is approximately 85 % at present, voting experiments with three statistical and one rule-based parser yield 87 % (cf. Zeman and Žabokrtský 2005), but these results also include an error rate of a statistical morphological disambiguation (tagger). With a better tagger the results of the parsers should also improve.

2.1 Exploting syntactic potentiality of disambiguation rules

2.1.1 Motivation

This section describes how disambiguation rules used in the process of the rule-based disambiguation can be extended to make shallow parsing possible. A large amount of very complex and very safe rules applied to an input text provides us with a lot of useful linguistic information about the sentence, but only a part of it has really been exploited during the disambiguation process up to now.

Many linguistic phenomena have been described by the rules for disambiguation purposes so far, for example a set of rules used for determining the borders of prepositional groups, a set of rules for identifying the antecedent of a relative pronoun, the rules for the unification of morphological categories of a verb with those of the corresponding subject etc. Some rules can be directly used not only for the disambiguation proper but also for providing overt information about syntactic relations in a sentence. In fact, a subset of (positive) rules is to be only slightly extended to provide syntactic information desired, and in order to capture syntactic phenomena not yet covered by the disambiguation rules one can even write rules intended specifically for parsing only (i.e. without performing disambiguation). This concerns also cases where a sentence has already been fully disambiguated and only syntactic links and chunks are to be established.

At present the disambiguation system contains about 2,000 rules, 430 of them were extended to provide syntactic information (particularly the positive ones, rather than the negative ones – see above; the rest of the rules does not provide any information usable at this stage or it provides information already contained in other rules). 80 additional rules were written for parsing purposes only. The work is still in progress and the number of rules of both types (disambiguation and syntactic) constantly increases.

As has been said above, there are basically two kinds of disambiguation rules: the negative ones and the positive ones. The negative rules, as the one presented in *Example 1*, are used for disambiguation only; the *positive ones can also be used for establishing syntactic relations*. Moreover, both kinds of rules, especially their configuration part can be used for *chunking*, i.e. for splitting an input sentence into relatively separate syntactic units. These three tasks, i.e.:

- disambiguation proper
- establishment of syntactic relations
- chunking

can be, as we shall see below, performed *in a parallel way*. More specifically, the positive approach can be used for *direct identification* of closely knit syntactic units such as:

- *clauses* within a compound sentence
- elements related by *grammatical concord* (in number, gender, case, person) in:
    - nominal groups
    - prepositional groups
    - analytical verbal predicates
    - subject-predicate relation
- various kinds of closely related adjacent syntagms (related e.g. by valency relations)
- *coordination structures* related by conjunctions/commas and by case identity
- contiguous *multi-word units* (processed by the collocation component).

2.1.2 Syntactic links

The specification of syntactic links will be demonstrated by an example of an extended disambiguation rule. In this example a simple positive disambiguation rule is presented that can be adapted to provide syntactic information. It is one of the bunch of rules designed to capture various kinds of subject-predicate agreement. In addition to the disambiguation proper, shallow parsing directive denoted *CONNECT* is included.

rule AgrSubjPred {

// configuration part starts

// beginning of the sentence
ITEM SentenceStart;

// syntactic adjectives in the subject nominal group
SEQUENCE OF IsSafe (SyntacticAdjective and Nominative);

// nominal subject in the nominative case *subject = ITEM IsSafe (Noun and Nominative);*

// elements standing between the subject and the verbal predicate in the same clause
SEQUENCE OF MustNotBe (Nominative or FiniteVerb or ClauseSeparator);

// verbal predicate, the copula *být* (E. *be*) is excluded
predicate = ITEM IsSafe (FiniteVerb and (not lemma == být"));

// executive part starts:

// disambiguation via unifying the predicate with the subject
UNIFY subject WITH predicate IN [person,number,gender];

// shallow parsing rule
CONNECT subject predicate Desc: Subj, Dom: Right;

}; // end of rule AgrSubjPred

Again, the executive part describes actions to be applied to any sentence which meets the conditions of the configuration part. The UNIFY function performs a disambiguation action which removes some tags (the ones which do not express agreement in the specified morphological categories) from the input words denoted *subject* and *predicate*[1]. The CONNECT directive, as an extension of the disambiguation proper, outputs syntactic information (it does not change any tags) as a syntactic link which can be used later by a parsing system for building a syntactic structure of the sentence.

2.1.3 Chunking

Disambiguation rules can also be extended in such a way as to include chunks, i.e. relatively independent closely knit syntactic structures as parts of an input sentence with a clear function in the sentence. Above all, the following types of structures should be identified as chunks:

- *contiguous* (parts of) *clauses* delimited by clause separators (punctuation marks, conjunctions)
- *nominal and prepositional groups*
- *coordinate structures* (difference between clausal and nonclausal coordination must be accounted for here)
- *analytical* (compound) *verbal predicates* (elements related by concord)
- *reflexive particles associated with their respective verbs/adjectives/nouns*
- *multi-word expressions*
- *verb-nominal expressions*.

The following disambiguation rule denoted *AgrInPrepGroup* will make it possible to delimit a chunk formed by a prepositional group.

Example 2. Chunk delimitation: prepositional group

[1] In Czech, the verbal predicate must agree with its subject represented mainly by a noun or pronoun in the nominative case in person, number and gender.

```
/* In Czech, elements in a prepositional group agree in case, number, gender */
rule AgrInPrepGroup {
ITEM ... /* left external boundary of the following prepositional group */
sfprep = ITEM IsSafe Preposition;
seq = SEQUENCE OF IsSafe (SyntacticAdjective or Adverb);
sfn = ITEM IsSafe Noun;
ITEM ... /* right external boundary of the preceding prepositional group */
INSERT sfprep INTO seq;
INSERT sfn INTO seq;
UNIFY CONDITIONALLY seq IN [case,number,gender];
};
```

Here the preposition group is formed by the elements:

```
sfprep = ITEM IsSafe Preposition;
seq = SEQUENCE OF IsSafe (SyntacticAdjective or Adverb);
sfn = ITEM IsSafe Noun;
```

if

- all nominal elements agree in number and gender and case
- all nominal elements are in the case required by the preposition.

This condition is expressed by the conditional unification command which unifies all the elements in the specified categories if there is a sequence of tags in the sentence being tagged which meets the agreement condition (adverbs are unifiable by default).

The elements of a prepositional group can form a chunk and in this case its left and right boundaries should be delimited. The disambiguation rule above can be extended so as to contain chunk identification:

```
/* chunk delimitation */

rule AgrInPrepGroup {
ITEM ... /* left external boundary of a prepositional group */
Create CHUNKBEG;
sfprep = ITEM IsSafe Preposition;
seq = SEQUENCE OF IsSafe (SyntacticAdjective or Adverb);
sfn = ITEM IsSafe Noun;
Create CHUNKEND;
ITEM ... /* right external boundary of the prepositional group */
INSERT sfprep INTO seq;
INSERT sfn INTO seq;
UNIFY CONDITIONALLY seq IN [case,number,gender];
};
```

The delimitation of the beginning and end of the chunk is performed by the *Create* commands but only in case the conditional unification is successful, i.e. if there is a sequence of elements meeting the agreement requirements specified.

Let us conclude this section by yet another simple example of chunk specification.

Example 3. Chunk delimitation: attribute-noun syntagm

```
/* Syntactically adjectival attribute */
rule AdjAttr {
ITEM IsSafe Verb; /*... left boundary ... */
Create CHUNKBEG;
adjat = ITEM (IsSafe SyntacticAdjective) and (MustNotBe AdjOptVal);
nounhead = ITEM IsSafe Noun;
Create CHUNKEND;
ITEM IsSafe (ClauseSeparator or Preposition or Verb);
 /*... right boundary ... */
UNIFY CONDITIONALLY adjat WITH nounhead IN [case,number,gender];
CONNECT adjat nounhead Desc: Attr, Dom: Right;
```

Here two words denoted *adjat* and *nounhead* form an attribute-noun syntagm in case they agree in number, gender and case and the syntactic adjective *adjat* has no syntactic valency. The fact that the syntactic adjective *adjat* has no syntactic valency is expressed by the *MustNotBe AdjOptVal* condition (if *adjat* had a certain valency, it would not be safe to establish the syntactic link specified because the adjective could be modified by the noun rather than vice versa). The chunk formed by the syntagm is also delimited.

2.2 The parsing process

The disambiguation system implemented in the LanGR programming language does not build any trees or other data structures. Morphological disambiguation is synchronized on data only and (in addition to chunking) syntactic information is produced at the output as a single CONNECT line consisting of the following items:

- identifiers of the words involved in the relation
- syntactic description of the relation
- information which of the two words is the governor (if applicable)
- optional information about the safety level of the rule (the omissible default is “very safe”).

For example, an application of the subject-predicate rule mentioned above to the sentence

(3) *Ema má mísu.*
E. Ema has a bowl.

produces the following output:

```
CONNECT 1 2 Desc: Subj, Dom: Right
```

Here the *Desc: Subj, Dom: Right* indicator means that item 1 depends on item 2 and that it is *Subj*ect of 2; from the word order viewpoint, the *Right* item is the governing one in the syntactic relation specified.

After the application of grammatical rules in LanGR which immediately morphologically disambiguate and produce syntactic information on the output a postprocessing phase comes. It is implemented as a Perl script which builds syntactic structures for the sentences by making use of and linking all syntactic information obtained from LanGR (and nothing else, neither the processed text itself, nor its morphology). The syntactic output from LanGR is not optimized for any syntactic formalism; in principle, it makes it possible to build structures in any suitable formalism (dependency structure, immediate constituents etc.). We have chosen dependency structures compatible with the analytic layer of the Prague Dependency Treebank (PDT) (cf. Hajič et al., 2001a) primarily for evaluation purposes, but we envisage to use our own formalism in future.

The linking process is not technical only, it also uses a number of linguistic facts. For example, if it is possible for a word to be either an attribute of a noun, or a member of a coordination, it becomes a member of the coordination and its parent (according to the definition of the analytic layer, it is a conjunction) "inherits" the possibility to be the attribute (of the same noun with the same safety level).

Let us illustrate this on the sentence

(4) *Jan*(1) *viděl*(2) *mladou*(3) *a*(4) *hezkou*(5) *dívku*(6).(7)
E. John(1) saw(2) a(_) young(3) and(4) pretty(5) girl(6).(7)

which obtains the following information from LanGR:

1. CONNECT 1 2 Desc: Subj, Dom: Right (*Jan* <u>*viděl*</u>)
2. CONNECT 2 0 Desc: AuxS, Dom: Right (*viděl* <u>SentenceBeg</u>)
3. CONNECT 3 4 Desc: Coord, Dom: Right (*mladou* <u>*a*</u>)
4. CONNECT 3 6 Desc: Attr, Dom: Right (*mladou* <u>*dívku*</u>)
5. CONNECT 4 6 Desc: Attr, Dom: Right (*a* <u>*dívku*</u>)
6. CONNECT 5 4 Desc: Coord, Dom: Right (*hezkou* <u>*a*</u>)
7. CONNECT 5 6 Desc: Attr, Dom: Right (*hezkou* <u>*dívku*</u>)
8. CONNECT 6 2 Desc: Obj, Dom: Right (*dívku* <u>*viděl*</u>)
9. CONNECT 7 0 Desc: AuxG, Dom: Right (. <u>SentenceBeg</u>)

Here the governor of each syntactic relation is underlined. SentenceBeg is an auxiliary symbol representing the whole sentence. In this case, the result of parsing the example sentence is a tree.

The third and the fifth word, i.e. *mladou* (young, 3) and *hezkou* (pretty, 5), respectively, has two possible parent nodes: the conjunction *a* (and, 4) and the noun *dívku* (girl, 6). We prefer the coordination (conjunction as the parent) and thus the conjunction overrides the second possible parent (noun). This preference generates new information:

CONNECT 4 6 Desc: Attr, Dom: Right

In this case, the information is redundant (we have obtained it already directly, see CONNECT line 5 above), but in other cases a new relation is established.

About thirty different transformations like this have been implemented in the postprocessing script. Some of them are more complicated; for example, if a relative pronoun is related to its antecent (in a coreferential relation) as well as to a verb (in a syntactic relation), it retains the syntactic relation with the verb and the verb becomes a child of the antecedent.

Thus, by applying extended disambiguation rules to an input sentence we obtain the following output:

- "very safe forest" (a group of syntactic trees representing parts of the sentence, or a single tree, in the ideal case) of partial trees is produced; in every tree syntactic relations are established
- chunks are specified.

Thus, disambiguation and shallow parsing go in parallel on two levels:

- morphological level (form)
- surface-syntactic level (function).

The forest of partial trees thus obtained can further be processed by a subsequent full-fledged parsing which would amalgamate all CONNECT-based and CHUNK-based syntactic pieces of information (partial trees) and form a resulting final tree if such a tree has not been achieved yet by the method described.

3. Conclusion

We have described a method of shallow parsing based on a very reliable disambiguation. The advantages of this approach for subsequent full parsing can be summarized as follows:

- the impact of disambiguation can be fully exploited
- chunking need not be blocked by part-of-speech and morphological ambiguities; just the opposite: it can make use of disambiguation
- forest of partial trees and chunks is produced to be subsequently used for composing a *final syntactic tree*
- rule-based disambiguation itself already detects many syntactic errors and in many cases it can identify erroneous sentences for which a subsequent parsing is at least very doubtful.

Bibliography

Charniak E. (2000): A Maximum-Entropy-Inspired Parser. In: Proceedings of NAACL. Seattle, Washington.

Collins M., J. Hajič, E. Brill, L. Ramshaw, Ch. Tillmann (1999): A Statistical Parser of Czech. In: Proceedings of the 37th Meeting of the ACL, 505–512. University of Maryland, College Park, Maryland.

Czech National Corpus. Faculty of Arts, Charles University, http://ucnk.ff.cuni.cz.

Hajič J., Hajičová E., Pajas P., Panevová J., Sgall P. and Vidová-Hladká B. (2001a): Prague dependency treebank, Prague. http://ufal.ms.mff.cuni.cz/pdt/

Hajič J., P. Krbec, P. Květoň, K. Oliva, V. Petkevič (2001b): Serial Combination of Rules and Statistics: A Case Study in Czech Tagging. In: Proceedings of the Conference of the 39th Annual Meeting of the Association for Computational Linguistics. CNRS – Institut de Recherche en Informatique de Toulouse and Université des Sciences Sociales, Toulouse, 260–267.

Hajič, J. (2004): Disambiguation of Rich Inflection (Computational Morphology of Czech), Vol. 1. Karolinum Charles University Press, Prague.

Karlsson F., A. Voutilainen, J. Heukkilä, A. Anttila (eds.) (1995): Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text. Mouton de Gruyter, Berlin and New York.

Květoň P. (2005): Rule-based Morphological Disambiguation. PhD Thesis, MFF, Charles University Prague.

Oliva K., M. Hnátková, V. Petkevič, P. Květoň (2000): The Linguistic Basis of a Rule-Based Tagger of Czech. Proceedings of the Text, Speech and Dialogue (TSD 2000) Third International Workshop. TSD 2000, LNAI 1902, Springer-Verlag Berlin Heidelberg, 3–8.

Oliva K. (2001): On Retaining Ambiguity in Disambiguated Corpora. In: Traitement Automatique des Langues vol. 42 No. 2, Hermes Science Publications, Paris.

Petkevič V. (2001): Automatic Detection of Subject and Verbal Predicate in the Czech Translation of G. Orwell's '1984'. In: Zybatow G., U. Junghanns, G. Mehlhorn, L. Szucsich (eds.): Current Issues in Formal Slavic Linguistics. Proceedings of the Third European Conference on Formal Description of Slavic Languages (FDSL'99) held in Leipzig. Peter Lang, Frankfurt am Main, 506–518.

Petkevič V. (2001): Grammatical Agreement and Automatic Morphological Disambiguation of Inflectional Languages. In: Matoušek V., P. Mautner, R. Mouček, K. Taušer (eds.): Proceedings of the Text, Speech and Dialogue conference TSD 2001 held in Železná Ruda, Czech Republic, 2001. TSD 2001, LNAI 2166, Springer-Verlag Berlin Heidelberg, 47–53.

Petkevič V., M. Hnátková (2002): Automatická morfologická disambiguace předložkových skupin v Českém národním korpusu. In Hladká Z., Karlík P. (eds.): ČEŠTINA – univerzália a specifika 4. Sborník konference v Brně, listopad 2001. Nakladatelství Lidové noviny, Praha, 243–252.

Petkevič V. (in press): Reliable Morphological Disambiguation of Czech: a Rule-Based Approach is Necessary. To appear in: M. Šimková (ed.): Insight into Slovak and Czech Corpus Linguistics, Veda 2005, Bratislava.

Votrubec J. (2005): Volba vhodných rysů pro morfologické značkování češtiny (Feature Selection for Morphological Tagging of Czech). Master thesis, MFF, Charles University, Prague.

Zeman D., Z. Žabortský (2005): Improving Parsing Accuracy by Combining Diverse Dependency Parsers. In: Proceedings of the International Workshop on Parsing Technologies (IWPT 2005). Simon Fraser University, Vancouver, British Columbia.