The Best of Two Worlds: Cooperation of Statistical and Rule-Based Taggers for Czech

Drahomíra "johanka" Spoustová
Jan Hajič
Jan Votrubec
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics,
Charles University Prague, Czech Republic
{johanka,hajic,votrubec}@ufal.mff.cuni.cz

Pavel Krbec
IBM Czech Republic,
Voice Technologies and Systems,
Prague, Czech Republic,
pavel_krbec@cz.ibm.com

Pavel Květoň
Institute of the Czech Language,
Academy of Sciences of the Czech Republic
Pavel.Kveton@seznam.cz

Abstract

Several hybrid disambiguation methods are described which combine the strength of hand-written disambiguation rules and statistical taggers. Three different statistical (HMM, Maximum-Entropy and Averaged Perceptron) taggers are used in a tagging experiment using Prague Dependency Treebank. The results of the hybrid systems are better than any other method tried for Czech tagging so far.

1 Introduction

Inflective languages pose a specific problem in tagging due to two phenomena: highly inflective nature (causing sparse data problem in any statistically based system), and free word order (causing fixed-context systems, such as n-gram HMMs, to be even less adequate than for English).

The average tagset contains about 1,000 – 2,000 distinct tags; the size of the set of possible and plausible tags can reach several thousands. There have been attempts at solving this problem for some of the highly inflective European languages, such as (Daelemans, 1996), (Erjavec, 1999) for Slovenian and (Hajič, 2000) for five Central and Eastern European languages.

Several taggers already exist for Czech, e.g. (Hajič et al., 2001b), (Smith, 2005), (Hajič et al., 2006) and (Votrubec, 2006). The last one reaches the best accuracy for Czech so far (95.12 %). Hence no system has reached – in the absolute terms – a performance comparable to English tagging (such as (Ratnaparkhi, 1996)), which stands above 97 %.

We are using the Prague Dependency Treebank (Hajič et al., 2006) (PDT) with about 1.8 million hand annotated tokens of Czech for training and testing. The tagging experiments in this paper all use the Czech morphological (pre)processor, which includes a guesser for "unknown" tokens and which is available from the PDT website (PDT Guide, 2006) to disambiguate only among those tags which are morphologically plausible.

The meaning of the Czech tags (each tag has 15 positions) we are using is explained in Table 1. The detailed linguistic description of the individual positions can be found in the documentation to the PDT (Hajič et al., 2006).

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 67–74,
Prague, June 2007.

	Name	Description
1	POS	Part of Speech
2	SUBPOS	Detailed POS
3	GENDER	Gender
4	NUMBER	Number
5	CASE	Case
6	POSSGENDER	Possessor's Gender
7	POSSNUMBER	Possessor's Number
8	PERSON	Person
9	TENSE	Tense
10	GRADE	Degree of comparison
11	NEGATION	Negation
12	VOICE	Voice
13	RESERVE1	Unused
14	RESERVE2	Unused
15	VAR	Variant

Table 1: Czech Morphology and the Positional Tags

2 Components of the hybrid system

2.1 The HMM tagger

The HMM tagger is based on the well known formula of HMM tagging:

$$\hat{T} = \arg\max_{T} P(T)P(W \mid T) \tag{1}$$

where

$$\begin{array}{l} P(W|T) \approx \prod_{i=1}^{n} P(w_i \mid t_i, t_{i-1}) \\ P(T) \approx \prod_{i=1}^{n} P(t_i \mid t_{i-1}, t_{i-2}). \end{array} \tag{2}$$

The trigram probability $P(W \mid T)$ in formula 2 replaces (Hajič et al., 2001b) the common (and less accurate) bigram approach. We will use this tagger as a baseline system for further improvements.

Initially, we change the formula 1 by introducing a scaling mechanism[1]: $\hat{T} = \arg\max_T(\lambda_T * logP(T) + logP(W \mid T))$.

We tag the word sequence from right to left, i.e. we change the trigram probability $P(W \mid T)$ from formula 2 to $P(w_i \mid t_i, t_{i+1})$.

Both the output probability $P(w_i \mid t_i, t_{i+1})$ and the transition probability $P(T)$ suffer a lot due to the data sparseness problem. We introduce a component $P(ending_i \mid t_i, t_{i+1})$, where $ending$ consists of the last three characters of w_i. Also, we introduce another component $P(t^*_i \mid t^*_{i+1}, t^*_{i+2})$ based on a reduced tagset T^* that contains positions POS, GENDER, NUMBER and CASE only (chosen on linguistic grounds).

[1]The optimum value of the scaling parameter λ_T can be tuned using held-out data.

We upgrade all trigrams to fourgrams; the smoothing mechanism for fourgrams is history-based bucketing (Krbec, 2005).

The final fine-tuned HMM tagger thus uses all the enhancements and every component contains its scaling factor which has been computed using held-out data. The total error rate reduction is 13.98 % relative on development data, measured against the baseline HMM tagger.

2.2 Morče

The Morče[2] tagger assumes some of the HMM properties at runtime, namely those that allow the Viterbi algorithm to be used to find the best tag sequence for a given text. However, the transition weights are not probabilities. They are estimated by an Averaged Perceptron described in (Collins, 2002). Averaged Perceptron works with features which describe the current tag and its context.

Features can be derived from any information we already have about the text. Every feature can be true or false in a given context, so we can regard current true features as a description of the current tag context.

For every feature, the Averaged Perceptron stores its weight coefficient, which is typically an integer number. The whole task of Averaged Perceptron is to sum all the coefficients of true features in a given context. The result is passed to the Viterbi algorithm as a transition weight for a given tag. Mathematically, we can rewrite it as:

$$w(C,T) = \sum_{i=1}^{n} \alpha_i.\phi_i(C,T) \tag{3}$$

where $w(C,T)$ is the transition weight for tag T in context C, n is number of features, α_i is the weight coefficient of i^{th} feature and $\phi(C,T)_i$ is evaluation of i^{th} feature for context C and tag T.

Weight coefficients (α) are estimated on training data, cf. (Votrubec, 2006). The training algorithm is very simple, therefore it can be quickly retrained and it gives a possibility to test many different sets of features (Votrubec, 2005). As a result, Morče gives the best accuracy from the standalone taggers.

[2]The name Morče stands for "MORfologie ČEštiny" ("Czech morphology").

2.3 The Feature-Based Tagger

The Feature-based tagger, taken also from the PDT (Hajič et al., 2006) distribution used in our experiments uses a general log-linear model in its basic formulation:

$$p_{AC}(y \mid x) = \frac{\exp(\sum_{i=1}^{n} \lambda_i f_i(y,x))}{Z(x)} \qquad (4)$$

where $f_i(y,x)$ is a binary-valued feature of the event value being predicted and its context, λ_i is a weight of the feature f_i, and the $Z(x)$ is the natural normalization factor.

The weights λ_i are approximated by Maximum Likelihood (using the feature counts relative to all feature contexts found), reducing the model essentially to Naive Bayes. The approximation is necessary due to the millions of the possible features which make the usual entropy maximization infeasible. The model makes heavy use of single-category Ambiguity Classes (AC)[3], which (being independent on the tagger's intermediate decisions) can be included in both left and right contexts of the features.

2.4 The rule-based component

The approach to tagging (understood as a stand-alone task) using hand-written disambiguation rules has been proposed and implemented for the first time in the form of Constraint-Based Grammars (Karlsson, 1995). On a larger scale, this aproach was applied to English, (Karlsson, 1995) and (Samuelsson, 1997), and French (Chanod, 1995). Also (Bick, 2000) uses manually written disambiguation rules for tagging Brazilian Portuguese, (Karlsson, 1985) and (Koskenniemi, 1990) for Finish and (Oflazer, 1997) reports the same for Turkish.

2.4.1 Overview

In the hybrid tagging system presented in this paper, the rule-based component is used to further reduce the ambiguity (the number of tags) of tokens in an input sentence, as output by the morphological processor (see Sect. 1). The core of the component is a hand-written *grammar* (set of rules).

Each rule represents a portion of knowledge of the language system (in particular, of Czech). The knowledge encoded in each rule is formally defined in two parts: a sequence of tokens that is searched for in an input sentence and the tags that can be deleted if the sequence of tokens is found.

The overall strategy of this "negative" grammar is to keep the highest recall possible (i.e. 100 %) and gradually improve precision. In other words, whenever a rule deletes a tag, it is (almost) 100% safe that the deleted tag is "incorrect" in the sentence, i.e. the tag cannot be present in any correct tagging of the sentence.

Such an (virtually) "error-free" grammar can partially disambiguate any input and prevent the subsequent taggers (stochastic, in our case) to choose tags that are "safely incorrect".

2.4.2 The rules

Formally, each rule consists of the description of the *context* (sequence of tokens with some special property), and the *action* to be performed given the context (which tags are to be discarded). The length of context is not limited by any constant; however, for practical purposes, the context cannot cross over sentence boundaries.

For example: in Czech, two finite verbs cannot appear within one clause. This fact can be used to define the following disambiguation rule:

- context: unambiguous finite verb, followed/preceded by a sequence of tokens containing neither a comma nor a coordinating conjunction, at either side of a word x ambiguous between a finite verb and another reading;
- action: delete the finite verb reading(s) at the word x.

It is obvious that no rule can contain knowledge of the whole language system. In particular, each rule is focused on at most a few special phenomena of the language. But whenever a rule deletes a tag from a sentence, the information about the sentence structure "increases". This can help other rules to be applied and to delete more and more tags.

For example, let's have an input sentence with two finite verbs within one clause, both of them ambiguous with some other (non-finite-verbal) tags. In this situation, the sample rule above cannot be applied.

[3] If a token can be a `N`(oun), `V`(erb) or `A`(djective), its (major POS) Ambiguity Class is the value "`ANV`".

On the other hand, if some other rule exists in the grammar that can delete non-finite-verbal tags from one of the tokens, then the way for application of the sample rule is opened.

The rules operate in a loop in which (theoretically) all rules are applied again whenever a rule deletes a tag in the partially disambiguated sentence. Since deletion is a monotonic operation, the algorithm is guaranteed to terminate; effective implementation has also been found in (Květoň, 2006).

2.4.3 Grammar used in tests

The grammar is being developed since 2000 as a standalone module that performs Czech morphological disambiguation. There are two ways of rule development:

- the rules developed by syntactic introspection: such rules are subsequently verified on the corpus material, then implemented and the implemented rules are tested on a testing corpus;
- the rules are derived from the corpus by introspection and subsequently implemented.

In particular, the rules are not based on examination of errors of stochastic taggers.

The set of rules is (manually) divided into two (disjoint) reliability classes — *safe* rules (100% reliable rules) and *heuristics* (highly reliable rules, but obscure exceptions can be found). The safe rules reflect general syntactic regularities of Czech; for instance, no word form in the nominative case can follow an unambiguous preposition. The less reliable heuristic rules can be exemplified by those accounting for some special intricate relations of grammatical agreement in Czech.

The grammar consists of 1727 safe rules and 504 heuristic rules. The system has been used in two ways:

- *safe rules only*: in this mode, safe rules are executed in the loop until some tags are being deleted. The system terminates as soon as no rule can delete any tag.
- *all rules*: safe rules are executed first (see *safe rules only* mode). Then heuristic rules start to operate in the loop (similarly to the safe rules). Any time a heuristic rule deletes a tag, the *safe rules only* mode is entered as a sub-procedure. When safe rules' execution terminates, the loop of heuristic rules continues. The disambiguation is finished when no heuristic rule can delete any tag.

The rules are written in the *fast LanGR* formalism (Květoň, 2006) which is a subset of more general LanGR formalism (Květoň, 2005). The LanGR formalism has been developed specially for writing and implementing disambiguation rules.

3 Methods of combination

3.1 Serial combination

The simplest way of combining a hand-written disambiguation grammar with a stochastic tagger is to let the grammar reduce the ambiguity of the tagger's input. Formally, an input text is processed as follows:

1. morphological analysis (every input token gets all tags that are plausible without looking at context);
2. rule-based component (partially disambiguates the input, i.e. deletes some tags);
3. the stochastic tagger (gets partially disambiguated text on its input).

This algorithm was already used in (Hajič et al., 2001b), only components were changed — the ruled-based component was significantly improved and two different sets of rules were tried, as well as three different statistical taggers. The best result was (not surprisingly) achieved with set of safe rules followed by the Morče tagger.

An identical approach was used in (Tapanainen, 1994) for English.

3.2 Serial combination with SUBPOS pre-processing

Manual inspection of the output of the application of the hand-written rules on the development data (as used in the serial combination described in the previous section) discovered that certain types of deadlocked ("cross-dependent") rules prevent successful disambiguation.

Cross-dependence means that a rule A can not apply because of some remaining ambiguity, which could be resolved by a rule B, but the operation of B is still dependent on the application of A. In particular, ambiguity in the Part-of-Speech category is very problematic. For example, only a few safe rules can apply to a three-word sentence where all three words are ambiguous between finite verbs and something else.

If the Part-of-Speech ambiguity of the input is already resolved, precision of the rule-based component and also of the final result after applying any of the statistical taggers improves. Full Part-of-Speech information is represented by the first two categories of the Czech morphology tagset — POS and SUBPOS, which deals with different types of pronouns, adverbs etc. As POS is uniquely determined by SUBPOS (Hajič et al., 2006), it is sufficient to resolve the SUBPOS ambiguity only.

All three taggers achieve more than 99% accuracy in SUBPOS disambiguation. For SUBPOS disambiguation, we use the taggers in usual way (i.e. they determine the whole tag) and then we put back all tags having the same SUBPOS as the tag chosen by the tagger.

Thus, the method with SUBPOS pre-processing operates in four steps:

1. morphological analysis;
2. SUBPOS disambiguation (any tagger);
3. rule-based component;
4. final disambiguation (the same tagger[4]).

The best results were again achieved with the tagger Morče and set of safe rules.

3.3 Combining more taggers in parallel

This method is quite different from previous ones, because it essentially needs more than one tagger. It consists of the following steps:

1. morphological analysis;
2. running N taggers independently;
3. merging the results from the previous step — each token ends up with between 1 and N tags, a union of the taggers' outputs;
4. (optional: the rule-based component;)
5. final disambiguation (single tagger).

The best results were achieved with two taggers in Step 1 (Feature-based and Morče), set of all rules in Step 3 and the HMM tagger in Step 4.

This method is based on an assumption that different stochastic taggers make complementary mistakes, so that the recall of the "union" of taggers is almost 100 %. Several existing language models are based on this assumption — (Brill, 1998) for tagging English, (Borin, 2000) for tagging German and (Vidová-Hladká, 2000) for tagging inflective languages. All these models perform some kind of "voting" — for every token, one tagger is selected as the most appropriate to supply the correct tag. The model presented in this paper, however, entrusts the selection of the correct tag to another tagger that already operates on the partially disambiguated input.

4 Results

All the methods presented in this paper have been trained and tested on the PDT version 2.0[5]. Taggers were trained on PDT 2.0 training data set (1,539,241 tokens), the results were achieved on PDT 2.0 evaluation-test data set (219,765 tokens), except Table 6, where PDT 2.0 development-test data set (201,651 tokens) was used. The morphological analysis processor and all the taggers were used in versions from April 2006 (Hajič et al., 2006), the rule-based component is from September 2006.

For evaluation, we use both precision and recall (and the corresponding F-measure) and accuracy, since we also want to evaluate the partial disambiguation achieved by the hand-written rules alone. Let t denote the number of tokens in the test data, let c denote the number of tags assigned to all tokens by a disambiguation process and let h denote

[4] This limitation is obviously not necessary, but we treat this combination primarily as a one-tagger method. Results of employing two different taggers are only slightly better, but still much worse than results of other methods presented later below.

[5] The results cannot be simply (number-to-number) compared to previous results on Czech tagging, because different training and testing data (PDT 2.0 instead of PDT 1.0) are used since 2006.

the number of tokens where the manually assigned tag is present in the output of the process.

- In case of the morphological analysis processor and the standalone rule-based component, the output can contain more than one tag for every token. Then *precision* (p), *recall* (r) and *F-measure* (f) characteristics are defined as follows:

$$p = h/c \qquad r = h/t \qquad f = 2pr/(p + r).$$

- The output of the stochastic taggers contains always exactly one tag for every token — then $p = r = f = h/t$ holds and this ratio is denoted as *accuracy*.

Table 2 shows the performance of the morphological analysis processor and the standalone rule-based component. Table 3 shows the performance of the standalone taggers. The improvement of the combination methods is presented in Table 4.

Table 5 shows the relative error rate reduction. The best method presented by this paper (parallel combination of taggers with all rules) reaches the relative error rate decrease of 11.48 % in comparison with the tagger Morče (which achieves the best results for Czech so far).

Table 6 shows error rate (100 % $-$ *accuracy*) of various methods[6] on particular positions of the tags (13 and 14 are omitted). The most problematic position is CASE (5), whose error rate was significantly reduced.

5 Conclusion

We have presented several variations of a novel method for combining statistical and hand-written rule-based tagging. In all cases, the rule-based component brings an improvement — the smaller the involvement of the statistical component(s) is, the bigger. The smallest gain can be observed in the case of the parallel combination of taggers (which by itself brings an expected improvement). The best variation improved the accuracy of the best-performing standalone statistical tagger by over 11 % (in terms of relative error rate reduction), and the inclusion of the rule-component itself improved the best statistical-only combination by over 3.5 % relative.

This might actually lead to pessimism regarding the rule-based component. Most other inflective languages however have much smaller datasets available than Czech has today; in those cases, we expect that the contribution of the rule-based component (which does not depend on the training data size, obviously) will be much more substantial.

The *LanGR* formalism, now well-developed, could be used for relatively fast development for other languages. We are, of course, unable to give exact figures of what will take less effort — whether to annotate more data or to develop the rule-based component for a particular language. Our feeling is that the jury is actually still out on this issue, despite some people saying that annotation is always cheaper: annotation for morphologically complex (e.g., inflective) languages is *not* cheap, and rule-based development efforts have not been previously using (unannotated) corpora so extensively (which is what *LanGR* supports for "testing" the developed rules, leading to more reliable rules and more effective development cycle).

On the other hand, the rule-based component has also two obvious and well-known disadvantages: it is language dependent, and the application of the rules is slower than even the baseline HMM tagger despite the "fast" version of the *LanGR* implementation we are using[7].

In any case, our experiments produced a software suite which gives the all-time best results in Czech tagging, and we have offered to apply it to re-tag the existing 200 mil. word Czech National Corpus. It should significantly improve the user experience (for searching the corpus) and allow for more precise experiments with parsing and other NLP applications that use that corpus.

[6] *F-b* stands for feature-based taggeer, *Par* for parallel combination without rules and *Par+Rul* for parallel combination with rules.

[7] In the tests presented in this paper, the speed of the operation of each stochastic tagger (and the parallel combination without rules) is several hundreds of tokens processed per second (running on a 2.2GHz Opteron processor). The operation of the standalone rule-based component, however, is cca 10 times slower — about 40 tokens per second. The parallel combination with all rules processes about 60 tokens per second — the rules operate faster here because their input in parallel combination is already partially disambiguated.

Method	p	r	f
Morphology	25.72 %	99.39 %	40.87 %
Safe rules	57.90 %	98.83 %	73.02 %
All rules	66.35 %	98.03 %	79.14 %

Table 2: Evaluation of rules alone

Tagger	accuracy
Feature-based	94.04 %
HMM	94.82 %
Morče	95.12 %

Table 3: Evaluation of the taggers alone

Combination method	accuracy
Serial (safe rules+Morče)	95.34 %
SUBPOS serial (safe rules+Morče)	95.44 %
Parallel without rules	95.52 %
Parallel with all rules	95.68 %

Table 4: Evaluation of the combinations

Method	Morče	Parallel without rules
Parallel without rules	8.20 %	—
Parallel with all rules	11.48 %	3.57 %

Table 5: Relative error rate reduction

	F-b	HMM	Morče	Par	Par+Rul
1	0.61	0.70	0.66	0.57	0.57
2	0.69	0.78	0.75	0.64	0.64
3	1.82	1.49	1.66	1.39	1.37
4	1.56	1.30	1.38	1.18	1.15
5	4.03	3.53	3.08	2.85	2.62
6	0.02	0.03	0.03	0.02	0.02
7	0.01	0.01	0.01	0.01	0.01
8	0.06	0.07	0.08	0.06	0.05
9	0.05	0.08	0.07	0.05	0.04
10	0.29	0.28	0.30	0.26	0.27
11	0.29	0.31	0.33	0.28	0.28
12	0.05	0.08	0.06	0.05	0.04
15	0.31	0.31	0.31	0.28	0.29

Table 6: Error rate [%] on particular positions of tags

Acknowledgements

The research described here was supported by the projects *MSM0021620838* and *LC536* of *Ministry of Eduation, Youth and Sports* of the Czech Republic, *GA405/06/0589* of the *Grant Agency* of the Czech Republic and *1ET100610409 Diagnostic and Evaluation Tools for Linguistic Software* of the *Information Society Programme* of the *National Research Programme* of the Czech Republic.

References

Eckhard Bick. 2000. The parsing system "Palavras" — automatic grammatical analysis of Portuguese in a constraint grammar framework. In: *Proceedings of the 2nd International Conference on Language Resources and Evaluation, TELRI.* Athens

Lars Borin. 2000. Something borrowed, something blue: Rule-based combination of POS taggers. In: *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, Vol. 1, pp. 21–26. Athens

Eric Brill and Jun Wu. 1998. Classifier combination for improved lexical disambiguation. In: *Proceedings of the 17th international conference on Computational linguistics*, Vol. 1, pp. 191–195. Montreal, Quebec

Jean-Pierre Chanod and Pasi Tapanainen. 1995. Tagging French — comparing a statistical and a constraint-based method. In: *Proceedings of EACL-95*, pp. 149–157. Dublin

Michael Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In: *Proceedings of EMNLP'02*, July 2002, pp. 1–8. Philadelphia

W. Daelemans and Jakub Zavrel and Peter Berck and Steven Gillis. 1996. MBT: A memory-based part of speech tagger-generator. In: *Proceedings of the 4th WVLC*, pp. 14–27. Copenhagen

Tomaz Erjavec and Saso Dzeroski and Jakub Zavrel. 1999. Morphosyntactic Tagging of Slovene: Evaluating PoS Taggers and Tagsets. *Technical Report*, Dept. for Intelligent Systems, Jozef Stefan Institute. Ljubljana

Jan Hajič and Barbora Hladká. 1997. Tagging of inflective languages: a comparison. In: *Proceedings of ANLP '97*, pp. 136–143. Washington, DC.

Jan Hajič 2000. Morphological tagging: Data vs. dictionaries. In: *Proceedings of the 6th ANLP / 1st NAACL'00*, pp. 94–101. Seattle, WA

Jan Hajič, Pavel Krbec, Pavel Květoň, Karel Oliva and Vladimír Petkevič. 2001. Serial Combination of Rules and Statistics: A Case Study in Czech Tagging. In: *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*. CNRS – Institut de Recherche en Informatique de Toulouse and Université des Sciences Sociales, pp. 260–267. Toulouse

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka and Marie Mikulová. 2006. Prague Dependency Treebank v2.0. CDROM. Linguistic Data Consortium, Cat. LDC2006T01. Philadelphia. ISBN 1-58563-370-4. Documentation also at `http://ufal.ms.mff.cuni.cz/pdt2.0`.

Fred Karlsson. 1985. Parsing Finnish in terms of a process grammar. In: Fred Karlsson (ed.): *Computational Morphosyntax: Report on Research 1981-84*, University of Helsinki, Department of General Linguistics Publications No. 13, pp. 137–176.

Fred Karlsson and Atro Voutilainen and Juha Heikkilä and Arto Anttila (eds.). 1995. Constraint Grammar: a language-independent system for parsing unrestricted text. *Natural Language Processing*. Vol. 4, Mouton de Gruyter, Berlin and New York.

Kimmo Koskenniemi. 1990. Finite-State Parsing and Disambiguation. In: *Proceedings of Coling-90*, University of Helsinki, 1990, pp. 229–232. Helsinki

Pavel Krbec. 2005. *Language Modelling for Speech Recognition of Czech.* PhD Thesis, MFF, Charles University Prague.

Pavel Květoň. 2005. *Rule-based Morphological Disambiguation.* PhD Thesis, MFF, Charles University Prague.

Pavel Květoň. 2006. Rule-based morphological disambiguation: On computational complexity of the LanGR formalism. In: *The Prague Bulletin of Mathematical Linguistics*, Vol. 85, pp. 57–72. Prague

Kemal Oflazer and Gökhan Tür. 1997. Morphological disambiguation by voting constraints. In: *Proceedings of the 8th conference on European chapter of the Association for Computational Linguistics*, pp. 222–229. Madrid

Karel Oliva, Milena Hnátková, Vladimír Petkevič and Pavel Květoň. 2000. The Linguistic Basis of a Rule-Based Tagger of Czech. In: Sojka P., Kopeček I., Pala K. (eds.): *Proceedings of the Conference "Text, Speech and Dialogue 2000", Lecture Notes in Artificial Intelligence*, Vol. 1902. Springer-Verlag, pp. 3–8. Berlin-Heidelberg

PDT Guide. `http://ufal.ms.mff.cuni.cz/pdt2.0`

A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In: *Proceedings of the 1st EMNLP*, May 1996, pp. 133–142. Philadelphia

Christer Samuelsson and Atro Voluntainen. 1997. Comparing a linguistic and a stochastic tagger. In: *Proceedings of ACL/EACL Joint Converence*, pp. 246–252. Madrid

Noah A. Smith and David A. Smith and Roy W. Tromble. 2005. Context-Based Morphological Disambiguation with Random Fields. In: *Proceedings of HLT/EMNLP*, pp. 475–482. Vancouver

Drahomíra "johanka" Spoustová. in prep. *Kombinované statisticko-pravidlové metody značkování češtiny. (Combining Statistical and Rule-Based Approaches to Morphological Tagging of Czech Texts).* PhD Thesis, MFF UK, in prep.

Pasi Tapanainen and Atro Voutilainen. 1994. Tagging accurately: don't guess if you know. In: *Proceedings of the 4th conference on Applied Natural Language Processing*, pp. 47–52. Stuttgart

Barbora Vidová-Hladká. 2000. *Czech Language Tagging.* PhD thesis, ÚFAL MFF UK. Prague

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

Jan Votrubec. 2006. Morphological Tagging Based on Averaged Perceptron. In: *WDS'06 Proceedings of Contributed Papers*, MFF UK, pp. 191–195. Prague