# Serial Combination of Rules and Statistics: A Case Study in Czech Tagging

Jan Hajič[1] Pavel Krbec[1] Pavel Květoň[2] Karel Oliva[3] Vladimír Petkevič[4]

[1]Institute of Formal and Applied Linguistics
MFF UK
Malostranské n. 25
118 00 Praha 1
Czech Republic
hajic@ufal.mff.cuni.cz, krbecp@cuni.cz

[2]Institute of Czech National Corpus
FF UK
n. Jana Palacha 2
116 35 Praha 1
Czech Republic
Pavel.Kveton@ff.cuni.cz

[3]Computational Linguistics
University of Saarland, Postfach 15 11 50
D-66 041 Saarbrücken
Germany
oliva@coli.uni-sb.de

[4]Institute of Theoretical and Computational Linguistics
FF UK
Celetná 13
110 00 Praha 1
Czech Republic
Vladimir.Petkevic@ff.cuni.cz

May 2, 2001

**Paper ID:** ACL-2001

**Keywords:** tagging, rule-based systems, statistical method, combination of methods, morphology

**Contact Author:** Jan Hajič, hajic@ufal.mff.cuni.cz

**Under consideration for other conferences (specify)?** none

## Abstract

A hybrid system is described which combines the strength of manual rule-writing and statistical learning, obtaining results superior to both methods if applied separately. The combination of a rule-based system and a statistical one is not parallel but serial: the rule-based system performing partial disambiguation with recall close to 100% is applied first, and a trigram HMM tagger runs on its results. An experiment in Czech tagging has been performed with encouraging results.

# Serial Combination of Rules and Statistics: A Case Study in Czech Tagging

**Abstract**

A hybrid system is described which combines the strength of manual rule-writing and statistical learning, obtaining results superior to both methods if applied separately. The combination of a rule-based system and a statistical one is not parallel but serial: the rule-based system performing partial disambiguation with recall close to 100% is applied first, and a trigram HMM tagger runs on its results. An experiment in Czech tagging has been performed with encouraging results.

## 1 Tagging of Inflective Languages

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 Hidden Markov Models (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.

Apart from agglutinative languages such as Turkish, Finnish and Hungarian (see e.g. (Hakkani-Tur et al., 2000)), and Basque (Ezeiza et al., 1998), which pose quite different and in the end less severe problems, there have been attempts at solving this problem for some of the highly inflectional European languages, such as (Daelemans et al., 1996), (Erjavec et al., 1999) (Slovenian), (Hajič and Hladká, 1997), (Hajič and Hladká, 1998) (Czech) and (Hajič, 2000) (five Central and Eastern European languages), but so far no system has reached - in the absolute terms - a performance comparable to English tagging (such as (Ratnaparkhi, 1996)), which stands around or above 97%. For example, (Hajič and Hladká, 1998) report results on Czech slightly above 93% only. One has to realize that even though such a performance might be adequate for some tasks (such as word sense disambiguation), for many other (such as parsing or translation) the implied sentence error rate at 50% (self-ref. omitted) or more is simply too much to deal with.

### 1.1 Statistical Tagging

Statistical tagging of inflective languages has been based on many techniques, ranging from plain-old HMM taggers (Mírovský, 1998), memory-based (Erjavec et al., 1999) to maximum-entropy and feature-based (Hajič and Hladká, 1998), (Hajič, 2000). For Czech, the best result achieved so far on approximately 300 thousand word training data set has been described in (Hajič and Hladká, 1998).

We are using 1.8M manually annotated tokens from the Prague Dependency Treebank (PDT) project (Hajič, 1998). We have decided to work with an HMM tagger[1] in the usual source-channel setting, with sophisticated smoothing. The HMM tagger uses the Czech morphological processor from PDT to disambiguate only among those tags which are morphologically plausible for a given input word form.

### 1.2 Manual Rule-based Systems

The idea of tagging by means of hand-written disambiguation rules has been put forward and implemented for the first time in the form of Constraint-Based Grammars (Karlsson et al., 1995). From languages we are acquainted with, the method has been applied on a larger scale only to English (Karlsson et al., 1995), (Samuelsson and Voutilainen, 1997), and French (Chanod and Tapanainen, 1995). Also (Bick, 1996) and (Bick, 2000) use manually written rules for Brazilian Portuguese, and there are several publications by Oflazer for Turkish.

Authors of such systems claim that hand-written systems can perform better than systems based on machine learning (Samuelsson and Voutilainen, 1997); however, except for the work cited, comparison is difficult to impossible due to the fact that they do not use the standard evaluation techniques (and not even the same data). But the substantial disadvantage is that the development of manual rule-based systems is demanding and requires a good deal of very subtle linguistic expertise and skills if full disambiguation also of “difficult” texts is to be performed. On the other hand, even very preliminary (initial) versions of these systems, while unable to disambiguate the difficult cases down to a single tag, almost never commit errors on an input which

[1] Mainly because of the ease with which it is trained even on large data, and also no other publicly available tagger was able to cope with the amount and ambiguity of the data in reasonable time.

can be considered linguistically "trivial".

We have decided to use the results achieved so far in writing a constraint-based grammar of Czech in combination with the current statistics-based tagger for Czech. This rule-based system also uses our Czech morphological analyzer as a preprocessor.

### 1.3 System Combination

Combination of (manual) rule-writing and statistical learning has been studied before. E.g., (Ngai and Yarowsky, 2000) and (Ngai, 2001) provide a thorough description of many experiments involving rule-based systems and statistical learners for NP bracketing. For tagging, combination of purely statistical classifiers has been described (Hladká, 2000), with about 3% relative improvement (error reduction from 18.6% to 18%, trained on small data) over the best original system. We regard such systems as working in parallel, since all the original classifiers run independently of each other.

In the present study, we have chosen a different strategy (similar to the one described for other types of languages in (Tapanainen and Voutilainen, 1994), (Ezeiza et al., 1998) and (Hakkani-Tur et al., 2000)). At the same time, the rule-based component is known to perform well in *eliminating* the incorrect alternatives[2], rather than picking the correct one under all circumstances. Moreover, the rule-based system used can examine the whole sentential context, again a difficult thing for a statistical system[3]. That way, the ambiguity of the input text[4] decreases. This is exactly what our statistical HMM tagger needs as its input, since it is already capable of using the lexical information from a dictionary to limit its lexical choices.

However, also in the rule-based approach, there is the usual tradeoff between precision and recall. We have decided to go for the "perfect" solution: to keep 100% recall, or very close to it, and gradually improve precision by writing rules which eliminate more and more incorrect tags. This way, we can be sure (or almost sure) that the performance of the HMM tagger performance will not be hurt by (recall) errors made by the rule component.

## 2 The Rule-based Component

### 2.1 Formal Means

Taken strictly formally, the rule-based component has the form of a restarting automaton with deletion (cf. (Plátek et al., 1995)), that is, each rule can be thought of as a finite-state automaton starting from the beginning of the sentence and passing to the right until it finds an input configuration on which it can operate by deletion of some parts of the input (the incorrect tags, in our case). Having performed this, the whole system is restarted, which means that the next rule is applied on the changed input (and this input is again read from the left end). This means that a single rule has the power of a finite state automaton, but the system as a whole has (even more than) a context-free power.

### 2.2 The Rules and Their Implementation

The system of hand-written rules for Czech has a twofold objective:

- practical: an error-free and at the same time the most accurate tagging of Czech texts
- theoretical: the description of the syntactic system of Czech, its *langue*, rather than *parole.*

The rules are to reduce the input ambiguity of the input text. During disambiguation the whole rule system combines two methods:

- the oblique one consisting in the elimination of syntactically wrong tag(s), i.e. in the reduction of the input ambiguity by deleting those tags which are excluded by the context
- the direct choice of the correct tag(s).

The overall strategy of the rule system is to keep the highest recall possible (i.e. 100 %) and gradually improve precision. Thus, the rules are assigned reliabilities (in %) which divide the rules into reliability classes, with the most reliable ("100%") group of rules applied first and less reliable groups of rules (threatening to decrease the 100% recall) being applied in subsequent steps. The 100% rules reflect general syntactic regularities of Czech; for instance, no word form in the nominative case can follow an unambiguous preposition. The less reliable rules can be exemplified by those accounting for some special intricate relations of grammatical agreement in Czech. Within each reliability group the rules are applied independently, i.e. in any order in a cyclic way until no ambiguity can be resolved.

Besides reliability, the rules can be generally divided according to the locality/nonlocality of their scope. Some phenomena (not many) in the structure of Czech sentence are local in nature: for instance, for the word "se" which is two-way ambiguous between a preposition (*with*) and a reflexive particle/pronoun (*himself*, as a particle) a prepositional

[2] Such a "negative" learning is thought to be difficult for any statistical system.

[3] Causing an immediate data sparseness problem.

[4] As prepared by the morphological analyzer.

reading can be available only in local contexts requiring the vocalisation of the basic form of the preposition “s” (*with*) resulting in the form “se”. However, in the majority of phenomena the correct disambiguation requires a much wider context. Thus, the rules use as wide context as possible with no context limitations being imposed in advance. During rules development performed so far, sentential context has been used, but nothing in principle limits the context to a single sentence. If it is generally appropriate for the disambiguation of the languages of the world to use unlimited context, it is especially fit for languages with free word order combined with rich inflection. There are many syntactic phenomena in Czech displaying the following property: a word form *wf1* can be part-of-speech determined by means of another word form *wf2* whose word-order distance cannot be determined by a fixed number of positions between the two word forms. This is exactly a general phenomenon which is grasped by the hand-written rules.

Formally, each rule consists of

- the description of the context (descriptive component), and
- the action to be performed given the context (executive component): i.e. which tags are to be discarded or which tag(s) are to be proclaimed correct (the rest being discarded as wrong).

For example,

- Context: not containing reflexive “se” and not containing any verb form of the lemma “dát”, “dávat”, “nechat”, “nechávat” on either side of the word “se” (lit. *himself*), followed by a verb from the list of all “reflexivum tantum” (i.e. verb requiring a “se” unconditionally)
- Action: at the word “se”, delete the prepositional reading

or

- Context: unambiguous finite verb, followed/preceded by a sequence of tokens containing neither comma nor 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 $x$.

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

The rules are formulated as generally as possible and at the same time as error-free (recall-wise) as possible. This approach of combining the requirements of maximum recall and maximum precision demands sophisticated syntactic knowledge of Czech. This knowledge is primarily based on the study of types of morphological ambiguity occurring in Czech. There are two main types of such ambiguity:

- regular (paradigm-internal)
- casual (lexical)

The regular (paradigm-internal) ambiguities occur within a paradigm, i.e. they are common to all lexemes belonging to a particular inflection class. This type of ambiguity can be exemplified by two examples: in Czech (as in many other inflective languages), the nominative, the accusative and the vocative case have the same form (in singular on the one hand, and in plural on the other); the same syncretism concerns nominative, accusative and vocative plural for all nominal paradigms except for masculine animate nouns. In Czech, the declension system is the most complex one, and the case syncretism of nominative and accusative in form can be considered the most difficult problem of the disambiguation of Czech (at least from the linguistic point of view).

The casual (lexical, paradigm-external) morphological ambiguity is lexically specific and hence cannot be investigated via paradigmatics. Nevertheless, a detailed knowledge of this ambiguity is essential for the practical aim of tagging. Thus more than 120 ambiguity classes have been developed (here a class is constituted by word forms displaying the same kind of ambiguity, e.g., the forms which have a reading as a noun in a particular case and number, and a verbal reading etc.), apart from hundreds of forms which are, as a rule, unique (the “class” has just one member) but display triple or more ambiguity.

In addition to the general rules, the whole rule approach includes a very important module which accounts for collocations and idioms. Their identification, description and classification constitutes one of the most crucial parts of the whole rule approach. The problem here consists in that the majority of collocations can – besides their most probable interpretation just as collocations – have also their literal meaning.

The rules are described in detail from the linguistic perspective in (self-reference omitted). Currently, the system (as evaluated in Sect. 2.3) consists of 80 rules.

At the beginning, the rules have been manually converted to a C++ code[5]. A new language for writing disambiguation rules is now being developed. The language is intended to be as powerful as general programming languages (in general we can talk about the power equaling that of the Turing machine). On the other hand, the syntax of the language will be very simple - thus it will be easily used by non-programmers, especially linguists.

The language is not a "grammar language" - the linguist writes rather a program than formal grammar rules. The program can be written either in plain language (we talk about "algoritmic rules", similar to C syntax, with the linguistic terms incorporated) or in a very descriptive configurational specification (we talk about "configurational rules" on surface level) - in this approach the linguist defines the main intrasentential context and some action to be performed if all specified context matches the input.

One of the main objectives of the language development is to remain I/O independent (e.g. to be able to work with any character encoding and any input formalism - XML, SGML and other) and to be flexibly configurable (mainly in the data structures area).

### 2.3 Evaluation of the Rule System Alone

The results are presented in Table 1. We use the usual equal-weight formula for F-measure:

$F\!-\!measure = \frac{2*Precision*Recall}{Precision+Recall}$,

where

$Precision = \frac{|\{Tokens\ with\ a\ correct\ tag\}|}{|\{Tokens\ generated\}|}$

and

$Recall = \frac{|\{Tokens\ with\ a\ correct\ tag\}|}{|\{Tokens\ in\ data\}|}$

## 3 The Statistical Component

### 3.1 The HMM Tagger

We have used an HMM tagger in the usual source-channel setting, fine-tuned to perfection using

- a 3-gram tag language model $p(t_i|t_{i-2}, t_{i-1})$,
- a tag-to-word lexical (translation) model using bigram histories instead of just same-word conditioning $p(w_i|t_i, t_{i-1})$[6],
- a bucketed linear interpolation smoothing for both models.

Thus the HMM tagger outputs a sequence of tags $T$ according to the usual equation

$T = argmax_T P(W|T)P(T)$,

where

$P(T) \approx \prod_{i=3..n} p_{smooth}(t_i|t_{i-2}, t_{i-1})$,

and

$P(W|T) \approx \prod_{i=3..n} p_{smooth}(w_i|t_i, t_{i-1})$.

Based on observations published recently in many papers that not using low frequency data hurts the overall performance, we do not discard any trigrams from the training data (i.e. we keep even singletons, which there are many - with more than 2000 different tags, it is no wonder).

The tagger has been trained in the usual way, using part of the training data as heldout data for smoothing of the two models employed. Smoothing has been done first without using buckets, and then with them to show the difference. Table 2 shows the resulting interpolation coefficients for the tag language model using the usual linear interpolation smoothing formula

$$p_{smooth}(t_i|t_{i-2}, t_{i-1}) = \\ \lambda_3 p(t_i|t_{i-2}, t_{i-1}) + \lambda_2 p(t_i|t_{i-1}) + \lambda_1 p(t_i) + \lambda_0/|V|$$

where p(...) is the "raw" Maximum Likelihood estimate of the probability distributions, i.e. the relative frequency in the training data.

The bucketing scheme for smoothing (a necessity when keeping all tag trigrams and tag-to-word bigrams) uses "buckets bounds" computed according to the following formula (for more on bucketing, see (Jelinek, 1997)):

$$v(h) = c(h)/|\{w : c(h, w) > 0\}|.$$

It should be noted that when using this bucketing scheme, the weights of the detailed distributions (with longest history) grow quickly as the history reliability increases. However, it is not monotonic; at several of the most reliable histories, the weight coefficients "jump" up and down. We have found that a sudden drop in $\lambda_3$ happens, e.g., for the bucket containing a history consisting of two consecutive punctuation symbols, which is not so much surprising after all.

[5] All the development is performed in the Linux OS environment.

[6] First used in (Thede and Harper, 1999), as far as we know.

| | Precision | Recall | F-measure ($\beta = 1$) |
|---|---|---|---|
| Morphology output only (baseline; no rules applied) | 28.97% | 100.00% | 44.92% |
| After application of manually written rules | 36.43% | 99.66% | 53.36% |

Table 1: Evaluation of rules alone, average on all 5 test sets

| | $\lambda_3$ | $\lambda_2$ | $\lambda_1$ | $\lambda_0$ |
|---|---|---|---|---|
| no buckets | 0.4371 | 0.5009 | 0.0600 | 0.0020 |
| bucket 0 (least reliable histories) | 0.0296 | 0.7894 | 0.1791 | 0.0019 |
| bucket 1 | 0.1351 | 0.7120 | 0.1498 | 0.0031 |
| bucket 2 | 0.2099 | 0.6474 | 0.1407 | 0.0019 |
| bucket 32 (most reliable histories) | 0.7538 | 0.2232 | 0.0224 | 0.0006 |

Table 2: Example smoothing coefficients for the tag language model (Exp 1 only)

A similar formula has been used for the lexical model (Table 3), and the strenghtening of the weights of the most detailed distributions has been observed, too.

### 3.2 Evaluation of the HMM Tagger alone

The HMM tagger described in the previous paragraph has achieved results shown in Table 4. It produces only the best tag sequence for every sentence, therefore only accuracy is reported (which is then equal to both precision and recall and F-measure, of course). Five-fold cross-validation has been performed (Exp 1-5) on a total data size of 1489983 tokens (excluding heldout data), divided up to five datasets of roughly the same size.

## 4 The Serial Combination

When the two systems are coupled together, the manual rules are run first, and then the HMM tagger runs as usual, except it selects from only those tags retained at individual tokens by the manual rule component, instead of from all tags as produced by the morphological analyzer:

- The morphological analyzer is run on the test data set. Every input token receives a list of possible tags based on an extensive Czech morphological dictionary.

- The manual rule component is run on the output of the morphology. The rules eliminate some tags which cannot form grammatical sentences in Czech.

- The HMM tagger is run on the output of the rule component, using only the remaining tags at every input token. The output is best-only; i.e., the tagger outputs exactly one tag per input token.

If there is no tag left at a given input token after the manual rules run, we reinsert all the tags from morphology and let the statistical tagger decide as if no rules had been used.

### 4.1 Evaluation of the Combined Tagger

Table 5 contains the final evaluation of the main contribution of this paper. Since the rule-based component does not attempt at full disambiguation, we can only use the F-measure for comparison and improvement evaluation[7].

The not-so-perfect recall of the rule component has been caused either by some deficiency in the rules, or by an error in the input morphology (due to a deficiency in the morphological dictionary), or by an error in the 'truth' (caused by an imperfect manual annotation). Since the number of recall errors is still substantially smaller than the overall number of errors, we do not investigate those errors here (in order not to compromise the test data by looking into them), thus we cannot offer any quantitative breakdown of them at this point[8]. Let's just lightly discuss the possible causes for a rule failure.

As Czech syntax is extremely complex, some of the rules are either not yet absolutely perfect, or they are too strict[9]. An example of the rule which decreases 100% recall for the test data is the following one:

In Czech, if an unambiguous preposition is detected in a clause, it "must" be followed - not nec-

[7] For the HMM tagger, which works in best-only mode, accuracy = precision = recall = F-measure, of course.

[8] We will do so when we get additional manually annotated data.

[9] "Too strict" is in fact good, given the overall scheme with the statistical tagger coming next, except in cases when it severely limits the possibility of increasing the precision. Nothing unexpected is happening here.

|  | $\lambda_3$ | $\lambda_2$ | $\lambda_1$ | $\lambda_0$ |
|---|---|---|---|---|
| no buckets | 0.3873 | 0.4461 | 0.0000 | 0.1666 |

Table 3: Example smoothing coefficients for the lexical model, no buckets (Exp 1 only)

|  | Accuracy (smoothing w/o bucketing) | Accuracy (bucketing) |
|---|---|---|
| Exp 1 | 95.23% | 95.34% |
| Exp 2 | 94.95% | 95.13% |
| Exp 3 | 95.04% | 95.19% |
| Exp 4 | 94.77% | 95.04% |
| Exp 5 | 94.86% | 95.11% |
| Average | 94.97% | 95.16% |

Table 4: Evaluation of the HMM tagger, 5-fold cross-validation

essarily immediately - by a nominal element (noun, adjective, pronoun or numeral) or, in very special cases, such a nominal element may be missing as it is elided. This fact about the syntax of prepositions in Czech is accounted for by a rule associating an unambiguous preposition with such a nominal element which is headed by the preposition (this rule is far from simple stating further conditions under which it can be applied; e.g., it deals with embedded adjectival phrases standing in between the preposition and the nominal element being headed by the preposition). The rule, however, erroneously ignores the fact that some prepositions function as heads of plain adverbs only (e.g., adverbs of time). As an example occurring in the test data we can take a simple structure "do kdy" (lit. *till when*), where "do" is a preposition (lit. *till*), *when* is an adverb of time and no nominal element follows. This results in the deletion of the prepositional interpretation of the preposition "do" thus causing an error. However, in cases like this, it is in fact easy to add another condition to the context (gaining back the lost recall) of such a rule rather than discard the rule as a whole (which would harm the precision too much).

### 4.2 Error Analysis

The main contribution of the architecture is that the combination of the systems does not commit linguistically trivial errors which from time to time occurred in the results of purely statistical tagging.

As examples of erroneous tagging results which have been eliminated for good due to the architecture described we might put forward:

- preposition requiring case $C$ not followed by any form in case $C$: Czech does not display essentially any form of preposition stranding, hence within a sentence, any preposition has to be followed by at least one form (of noun, adjective, pronoun or numeral) in the case required. Turning this around, if a word which is ambiguous between a preposition and another part of speech is not followed by the respective form till the end of the sentence, it is safe to discard the prepositional reading in almost all non-idiomatic, non-coordinated cases.
- two finite verbs within a clause: Similarly to most of other languages, a Czech clause must not contain more than one finite verb. This means that if two words, one genuine finite verb and the other one ambiguous between a finite verb and another reading, stand in such a configuration that the material between them contains no clause separator (comma, conjunction), it is safe to discard the finite verb reading with the ambiguous word.
- two nominative cases within a clause: The subject in Czech is usually case-marked by nominative, and simultaneously, even when the position of subject is free (it can stand both to the left or to the right of the main verb) in Czech, no clause can have two non-coordinated subjects; hence, if there are no special reasons (which can be put into the rules) for two nominatives to occur within a clause, the ambiguity can be reduced in case there occur two nominative nouns within a clause, one genuine and one ambiguous with another reading.

## 5 Conclusions

The improvements obtained (4.58% relative error reduction) beat the pure statistical classifier combination (Hladká, 2000), which obtained only 3% relative improvement. The most important task for the

|  | HMM (w/bucketing) | Rules | Combined | diff. combined - HMM (rel.) |
|---|---|---|---|---|
| Exp 1 | 95.34% | 53.65% | 95.53% | 4.08% |
| Exp 2 | 95.13% | 52.39% | 95.36% | 4.72% |
| Exp 3 | 95.19% | 53.49% | 95.41% | 4.57% |
| Exp 4 | 95.04% | 53.44% | 95.28% | 4.84% |
| Exp 5 | 95.11% | 53.82% | 95.34% | 4.70% |
| Average | 95.16% | 53.36% | 95.38% | 4.58% |

Table 5: F-measure-based evaluation of the combined tagger, 5-fold cross-validation

| Word Form | Annotator | Tagger |
|---|---|---|
| Malé (*Small*) | `AAFP1----1A----` | `AAFP1----1A----` |
| organizace (*businesses*) | `NNFP1-----A----` | `NNFP1-----A----` |
| mají (*have*) | `VB-P---3P-AA---` | `VB-P---3P-AA---` |
| problémy (*problems*) | `NNIP4-----A----` | `NNIP4-----A----` |
| **se** *(with)*..............(!ERROR!) | `P7-X4----------` | `RV--7----------` |
| získáním (*getting*) | `NNNS7-----A----` | `NNNS7-----A----` |
| telefonních (*phone*) | `AAFP2----1A----` | `AAFP2----1A----` |
| linek (*lines*) | `NNFP2-----A----` | `NNFP2-----A----` |

Figure 1: Annotation error: `P7-X4----------`, refl. pronoun tag should be a preposition `RV--7----------`

manual-rule component is to keep recall very close to 100%, with the task of improving precision as much as possible. Even though the rule-based component is still under development[10], the 19% relative improvement in F-measure over the baseline (i.e., 16% reduction in the F-complement while keeping recall just 0.34% under the absolute one) is encouraging.

In any case, we consider the clear "division of labor" between the two parts of the system a strong advantage. It allows now and in the future to use different taggers and different rule-based systems within the same framework but in a completely independent fashion.

The performance of the pure HMM tagger alone is an interesting result by itself, beating the best Czech tagger published (Hajič and Hladká, 1998) by almost 2% (30% relative improvement) and a previous HMM tagger on Czech (Mírovský, 1998) by almost 4% (44% relative improvement). We believe that the key to this success is both the increased data size (we have used three times more training data then reported in the previous papers) and the meticulous implementation of smoothing with bucketing together with using all possible tag trigrams, which has never been done before.

One might question whether it is worthwhile to work on a manual rule component if the improvement over the pure statistical system is not so huge, and there is the obvious disadvantage in its language-specificity. However, we see at least two situations in which this is the case: first, the need for high quality tagging for local language projects, such as human-oriented lexicography, where every 1/10th of a percent of reduction in error rate counts, and second, a situation where not enough training data is available for a high-quality statistical tagger for a given language, but a language expertise does exist; the improvement over an imperfect statistical tagger should then be more visible[11].

Another interesting issue is the evaluation method used for taggers. From the linguistic point of view, not all errors are created equal; it is clear that the manual rule component does not commit linguistically trivial errors (see Sect. 4.2). However, the relative weighting (if any) of errors should be application-based, which is already outside of the scope of this paper.

It has been also observed that the improved tagger can serve as an additional means for discovering annotator's errors (however infrequent they are, they are there). See Fig. 1 for an example of wrong annotation of "se".

In the near future, we plan to add more rules, as well as continue to work on the statistical tagging.

[10] for reviewers: expected to include still better results in the final version, since further rules are still being added.

[11] However, a feature-based log-linear tagger might perform better for small training data, as argued in (Hajič, 2000).

The lexical component of the tagger might still have some room for improvement, such as the use of

$$P(W|T) \approx \prod_{i=3..n} p_{smooth}(w_i|t_i, w_{i-1}),$$

which can be feasible with the powerfull smoothing we now employ.

## 6 Acknowledgements

Excluded in the submission to avoid self-reference.

## References

E. Bick. 1996. Automatic parsing of Portuguese. *Proceedings of the Second Workshop on Computational Processing of Written Portuguese, Curitiba*, pages 91–100.

E. Bick. 2000. The parsing system "Palavras" - automatic grammatical analysis of Portuguese in a constraint grammar framework. *2nd International Conference on Language Resources and Evaluation, Athens, Greece.*

J. P. Chanod and P. Tapanainen. 1995. Tagging French - comparing a statistical and a constraint-based method. In *Proceeedings of EACL-95*, pages 149–157, Dublin.

Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. MBT: A memory-based part of speech tagger generator. In *Proceedings of WVLC 4*, pages 14–27. ACL.

Tomaž Erjavec, Saso Dźeroski, and Jakub Zavrel. 1999. Morphosyntactic Tagging of Slovene: Evaluating PoS Taggers and Tagsets. Technical Report IJS-DP 8018, Dept. for Intelligent Systems, Józef Štefan Institute, Ljubljana, Slovenia, April 2nd.

N. Ezeiza, I. Alegria, J. M. Ariola, R. Urizar, and I. Aduriz. 1998. Combining stochastic and rule-based methods for disambiguation in agglutinative languages. In *Proceedings of the 36th Annual Meeting of the ACL and 17th Coling 1998*, Montreal, Canada.

Jan Hajič. 1998. Building a syntactically annotated corpus: The Prague Dependency Treebank. In E. Hajičová, editor, *Festschrift for Jarmila Panevová*, pages 106–132. Karolinum, Charles University, Prague.

Jan Hajič. 2000. Morphological tagging: Data vs. dictionaries. In *Proceedings of the NAACL'00*, pages 94–101, Seattle, WA, May. ACL.

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

Jan Hajič and Barbora Hladká. 1998. Tagging inflective languages: Prediction of morphological categories for a rich, structured tagset. In *Proceedings of ACL/COLING'98*, pages 483–490, Montreal, Canada. ACL/ICCL.

D. Hakkani-Tur, K. Oflazer, and G. Tur. 2000. Statistical morphological disambiguation for agglutinative languages. In *Proceedings of the 18th Coling 2000*, Saarbruecken, Germany.

Barbora Hladká. 2000. *Czech Language Tagging.* Ph.D. thesis, ÚFAL, Faculty of Mathematics and Physics, Charles University, Prague. 135 pp.

Fred Jelinek. 1997. *Statistical Methods for Speech Recognition.* MIT Press, Cambridge, MA.

F. Karlsson, A. Voutilainen, J. Heikkilä, and A. Antilla, editors. 1995. *Constraint Grammar: a Language-Independent System for Parsing Unrestricted Text.* Mouton de Gruyter, Berlin New York.

Jiří Mírovský. 1998. Morfologické značkování textu: automatická disambiguace (in Czech). Master's thesis, ÚFAL, Faculty of Mathematics and Physics, Charles University, Prague. 56 pp.

G. Ngai and D. Yarowsky. 2000. Rule writing or annotation: Cost-efficient resource usage for base noun phrase chunking. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 117–125, Hong Kong.

G. Ngai. 2001. *Maximizing Resources for Corpus-Based Natural Language Processing.* Ph.D. thesis, Johns Hopkins University, Baltimore, Maryland, USA, January.

M. Plátek, P. Jančar, F. Mráz, and J. Vogel. 1995. On restarting automata with rewriting. Technical Report 96/5, Charles University, Prague.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP 1*, pages 133–142. ACL.

C. Samuelsson and A. Voutilainen. 1997. Comparing a linguistic and a stochastic tagger. In *Proceedings of ACL/EACL Joint Conference*, pages 246–252, Madrid.

P. Tapanainen and A. Voutilainen. 1994. Tagging accurately: Don't guess if you know. Technical report, Xerox Corp.

Scott M. Thede and Mary P. Harper. 1999. A Second-Order Hidden Markov Model for Part-of-Speech Tagging. *Proceedings of ACL'99*, pages 175–182.