Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset

Jan Hajič and **Barbora Hladká**
Institute of Formal and Applied Linguistics MFF UK
Charles University, Prague, Czech Republic
{hajic,hladka}@ufal.mff.cuni.cz

Abstrakt (česky)

{This short abstract is in Czech. For illustration purposes, it has been tagged by our tagger; errors are printed underlined and corrections are shown.}

Hlavním/AAIS7----1A--
problémem/NNIS7-----A--
při/RR--6--------
morfologickém/AANS6----1A--
značkování/NNNS6-----A--
(/Z:-----------
někdy/Db-----------
též/Db-----------
zvaném/AAIS6----1A-- Correct: N
morfologicko/A2-----------
-/Z:-----------
syntaktické/AAIP1----1A-- Correct: NS
)/Z:-----------
jazyků/NNIP2-----A--
s/RR--7--------
bohatou/AAFS7----1A--
flexí/NNFS7-----A--
,/Z:-----------
jako/J,-----------
je/VB-S---3P-AA-
například/Db-----------
čeština/NNFS1-----A--
nebo/J^-----------
ruština/NNFS1-----A--
,/Z:-----------
je/VB-S---3P-AA-
-/Z:-----------
při/RR--6--------
omezené/AAFS6----1A--
velikosti/NNFS2-----A-- Correct: 6
zdrojů/NNIP2-----A--
-/Z:-----------
počet/NNIS1-----A--
možných/AAFP2----1A--
značek/NNFP2-----A--
,/Z:-----------
který/P4YS1--------
jde/VB-S---3P-AA-
obvykle/Dg-------1A--
do/RR--2--------
tisíců/NNIP2-----A--
./Z:-----------
Naše/PSHS1-P1-----
metoda/NNFS1-----A--
přitom/Db-----------
využívá/VB-S---3P-AA-
exponenciálního/AAIS2----1A--
pravděpodobnostního/AAIS2----1A--
modelu/NNIS2-----A--
založeného/AAIS2----1A--
na/RR--6--------
automaticky/Dg-------1A--
vybraných/AAMP6----1A-- Correct: I
rysech/NNIP6-----A--
./Z:-----------
Parametry/NNIP1-----A--
tohoto/PDZS2--------
modelu/NNIS2-----A--
se/P7-X4--------
počítají/VB-P---3P-AA-
pomocí/NNFS7-----A-- Correct: RR--2,-
jednoduchých/AAIP2----1A--
odhadů/NNIP2-----A--
(/Z:-----------
trénink/NNIS1-----A--
je/VB-S---3P-AA-
tak/Db-----------
mnohem/Db-----------
rychlejší/AAFS1----2A-- Correct: I
,/Z:-----------
než/J,-----------
kdybychom/J,-P---1-----
použili/VpMP---XR-AA-
metodu/NNFS4-----A--
maximální/AAFS4----1A-- Correct: 2
entropie/NNFS2-----A--
)/Z:-----------
a/J^-----------
přitom/Db-----------
se/P7-X4--------
přímo/Dg-------1A--
minimalizuje/VB-S---3P-AA-
počet/NNIS4-----A-- Correct: 1
chyb/NNFP2-----A--
./Z:-----------

Abstract

The major obstacle in morphological (sometimes called morpho-syntactic, or extended POS) tagging of highly inflective languages, such as Czech or Russian, is – given the resources possibly available – the tagset size. Typically, it is in the order of thousands. Our method uses an exponential probabilistic model based on automatically selected features. The parameters of the model are computed using simple estimates (which makes training much faster than when one uses Maximum Entropy) to directly minimize the error rate on training data.

The results obtained so far not only show good performance on disambiguation of most of the individual morphological categories, but they also show a significant improvement on the overall prediction of the resulting combined tag over a HMM-based tag n-gram model, using even substantially less training data.

1 Introduction

1.1 Orthogonality of morphological categories of inflective languages

The major obstacle in morphological[1] tagging of highly inflective languages, such as Czech or Russian, is – given the resources possibly available – the tagset size. Typically, it is in the order of thousands. This is due to the (partial) "orthogonality"[2] of simple morphological categories, which then multiply when creating a "flat" list of tags. However, the individual categories contain only a very small number of different values; e.g., number has five (Sg, Pl, Dual, Any, and "not applicable"), case nine etc.

The "orthogonality" should not be taken to mean complete independence, though. Inflectional languages (as opposed to agglutinative languages such as Finnish or Hungarian) typically combine several certain categories into one morpheme (suffix or ending). At the same time, the morphemes display a high degree of ambiguity, even across major POS categories.

For example, most of the Czech nouns can form singular and plural forms in all seven cases, most adjectives can (at least potentially) form all (4) genders, both numbers, all (7) cases, all (3) degrees of comparison, and can be either of positive or negative polarity. That gives 336 possibilities (for adjectives), many of them homonymous on the surface. On the other hand, pronouns and numerals do not display such an orthogonality, and even adjectives are not fully orthogonal – an ancient "dual" number, happily living in modern Czech in the feminine, plural and instrumental case adds another 6 sub-orthogonal possibilities to almost every adjective. Together, we employ 3127 plausible combinations (including style and diachronic variants).

1.2 The individual categories

There are 13 morphological categories currently used for morphological tagging of Czech: part of speech, detailed POS (called "subpart of speech"), gender, number, case, possessor's gender, possessor's number, person, tense, degree of comparison, negativeness (affirmative/negative), voice (active/passive), and variant/register.

The `POS` category contains only the major part of speech values (noun (`N`), verb (`V`), adjective (`A`), pronoun (`P`), verb (`V`), adjective (`A`), adverb (`D`), numeral (`C`), preposition (`R`), conjunction (`J`), interjection (`I`), particle (`T`), punctuation (`Z`), and "undefined" (`X`)). The "subpart of speech" (`SUBPOS`) contains details about the major category and has 75 different values. For example, verbs (`POS`: `V`) are divided into simple finite form in present or future tense (`B`), conditional (`c`), infinitive (`f`), imperative (`i`), etc.[3]

All the categories vary in their size as well as in their unigram entropy (see Table 1) computed using the standard entropy definition

$$H_p = -\sum_{y \in Y} p(y) log(p(y)) \tag{1}$$

where p is the unigram distribution estimate based on the training data, and Y is the set of possible values of the category in question. This formula can be rewritten as

$$H_{p,D} = -\frac{1}{|D|}\sum_{i=1}^{|D|} log(p(y_i)) \tag{2}$$

where p is the unigram distribution, D is the data and $|D|$ its size, and y_i is the value of the category in question at the $i-th$ event (or position) in the data. The form (2) is usually used for cross-entropy computation on data (such as test data) different from those used for estimating p. The base of the log function is always taken to be 2.

1.3 The morphological analyzer

Given the nature of inflectional languages, which can generate many (sometimes thousands of) forms for a given lemma (or "dictionary entry"), it is necessary to employ morphological analysis before the tagging proper. In Czech, there are as many as 5 different lemmas (not counting underlying derivations nor

[1] This type of tagging is sometimes called morpho-syntactic tagging. However, to stress that we are not dealing with syntactic categories such as Object or Attribute (but rather with morphological categories such as Number or Case) we will use the term "morphological" here.

[2] By orthogonality we mean that all combinations of values of two (or more) categories are systematically possible, i.e. that every member of the cartesian product of the two (or more) sets of values do appear in the language.

[3] The categories `POS` and `SUBPOS` are the only two categories which are rather lexically (and not inflectionally) based.

Table 1: Most Difficult Individual Morphological Categories

Category	Number of values	Unigram entropy H_p (in bits)
POS	12	2.99
SUBPOS	75	3.83
GENDER	11	2.05
NUMBER	6	1.62
CASE	9	2.24
POSSGENDER	5	0.04
POSSNUMBER	3	0.04
PERSON	5	0.64
TENSE	6	0.55
GRADE	4	0.55
NEGATION	3	1.07
VOICE	3	0.45
VAR	10	0.07

word senses) and up to 108 different tags for an input word form. The morphological analyzer used for this purpose (Hajič, in prep.), (Hajič, 1994) covers about 98% of running unrestricted text (newspaper, magazines, novels, etc.). It is based on a lexicon containing about 228,000 lemmas and it can analyze about 20,000,000 word forms.

2 The Training Data

Our training data consists of about 130,000 tokens of newspaper and magazine text, manually double-tagged and then corrected by a single judge.

Our training data consists of about 130,000 tokens of newspaper and magazine text, manually tagged using a special-purpose tool which allows for easy disambiguation of morphological output. The data has been tagged twice, with manual resolution of discrepancies (the discrepancy rate being about 5%, most of them being simple tagging errors rather than opinion differences).

One data item contains several fields: the input word form (token), the disambiguated tag, the set of all possible tags for the input word form, the disambiguated lemma, and the set of all possible lemmas with links to their possible tags. Out of these, we are currently interested in the form, its possible tags and the disambiguated tag. The lemmas are ignored for tagging purposes.[4]

The tag from the "disambiguated tag" field as well as the tags from the "possible tags" field are further divided into so called **subtags** (by morphological category). In the set "possible tags field", the ambiguity on the level of full (combined) tags is mapped onto so called "ambiguity classes" (*AC*-s) of subtags. This mapping is generally not reversible, which means that the links across categories might not be preserved. For example, the word form *jen* for which the morphology generates three possible tags, namely, `TT-----------` (particle "only"), and `NNIS1-----A--` and `NNIS4-----A--` (noun, masc. inanimate, singular, nominative (1) or accusative (4) case; "yen" (the Japanese currency)), will be assigned six ambiguous ambiguity classes (`NT, NT, -I, -S, -14, -A`, for POS, subpart of speech, gender, number, case, and negation) and 7 unambiguous ambiguity classes (all `-`). An example of the training data is presented in Fig. 1. It contains three columns, separated by the vertical bar (|):

```
RR--6--------|R/R/-/-/46/-/-/-/-/-/-/-/-/|na
AAIS6----1A--|A/A/IMN/S/6/-/-/-/-/1/A/-/-/|počítačovém
NNIS6-----A--|N/N/I/S/236/-/-/-/-/-/A/-/-/|modelu
Z:-----------|Z/:/-/-/-/-/-/-/-/-/-/-/-/|,
P4YS1--------|P/4/IY/S/14/-/-/-/-/-/-/-/-/|který
VpYS---XR-AA-|V/p/Y/S/-/-/-/X/R/-/A/-/-/|simuloval
NNIS4-----A--|N/N/I/S/14/-/-/-/-/-/A/-/-/|vývoj
AANS2----1A--|A/A/IMN/S/24/-/-/-/-/1/A/-/-/|světového
NNNS2-----A--|N/N/N/S/236/-/-/-/-/-/A/-/-/|klimatu
RR--6--------|R/R/-/-/46/-/-/-/-/-/-/-/-3/|v
AANP6----1A--|A/A/FIMN/P/26/-/-/-/-/1/A/-/-/|příštích
NNNP6-----A--|N/N/N/P/6/-/-/-/-/-/A/-/-/|desetiletích
```

Figure 1: Training Data: lit: *on computer(adj.) model , which was-simulating development of-world climate in next decades*

1. the "truth" (the correct tag, i.e. a sequence of 13 subtags, each represented by a single character, which is the true value for each individual category in the order defined in Fig. 1 (1st column: POS, 2nd: SUBPOS, etc.)
2. the 13-tuple of ambiguity classes, separated by a slash (/), in the same order; each ambiguity class is named using the single character subtags used for all the possible values of that category;
3. the original word form.

Please note that it is customary to number the seven grammatical cases in Czech: (instead of naming them): "nominative" gets `1`, "genitive" `2`, etc. There are four genders, as the Czech masculine gender is divided into masculine animate (`M`) and inanimate (`I`).

Fig. 1 is a typical example of the ambiguities encountered in a running text: little POS ambiguity, but a lot of gender, number and case ambiguity (columns 3 to 5).

[4] In fact, tagging helps in most cases to disambiguate the lemmas. Lemma disambiguation is a separate process following tagging. The lemma disambiguation is a much simpler problem – the average number of different lemmas per token (as output by the morphological analyzer) is only 1.15. We do not cover the lemma disambiguation procedure here.

3 The Model

Instead of employing the source-channel paradigm for tagging (more or less explicitly present e.g. in (Merialdo, 1992), (Church, 1988), (Hajič, Hladká, 1997)) used in the past (notwithstanding some exceptions, such as Maximum Entropy and rule-based taggers), we are using here a "direct" approach to modeling, for which we have chosen an exponential probabilistic model. Such model (when predicting an event[5] $y \in Y$ in a context x) has the general form

$$p_{AC,e}(y|x) = \frac{\exp(\sum_{i=1}^{n} \lambda_i f_i(y,x))}{Z(x)} \tag{3}$$

where $f_i(y,x)$ is the set (of size n) of binary-valued (yes/no) *features* of the event value being predicted and its context, λ_i is a "weigth" (in the exponential sense) of the feature f_i, and the normalization factor $Z(x)$ is defined naturally as

$$Z(x) = \sum_{y \in Y} \exp(\sum_{i=1}^{n} \lambda_i f_i(y,x)) \tag{4}$$

We use a separate model for each ambiguity class AC (which actually appeared in the training data) of each of the 13 morphological categories[6]. The final $p_{AC}(y|x)$ distribution is further smoothed using unigram distributions on subtags (again, separately for each category).

$$p_{AC}(y|x) = \sigma p_{AC,e}(y|x) + (1-\sigma) p_{AC,1}(y) \tag{5}$$

Such smoothing takes care of any unseen context; for ambiguity classes not seen in the training data, for which there is no model, we use unigram probabilities of subtags, one distribution per category.

In the general case, features can operate on any imaginable context (such as the speed of the wind over Mt. Washington, the last word of yesterday TV news, or the absence of a noun in the next 1000 words, etc.). In practice, we view the context as a set of attribute-value pairs with a discrete range of values (from now on, we will use the word "context" for such a set). Every feature can thus be represented by a set of contexts, in which it is positive. There is, of course, also a distinguished attribute for the value of the variable being predicted (y); the rest of the attributes is denoted by x as expected. Values of attributes will be denoted by an overstrike ($\overline{y}, \overline{x}$).

The pool of contexts of prospective features is for the purpose of morphological tagging defined as a full cross-product of the category being predicted (y) and of the x specified as a combination of:

1. an ambiguity class of a single category, which may be different from the category being predicted, or
2. a word form

and

1. the current position, or
2. immediately preceding (following) position in text, or
3. closest preceding (following) position (up to four positions away) having a certain ambiguity class in the POS category

Let now

> *Categories* = {*POS, SUBPOS, GENDER, NUMBER, CASE, POSSGENDER, POSSNUMBER, PERSON, TENSE, GRADE, NEGATION, VOICE, VAR*};

then the feature function $f_{Cat_{AC},\overline{y},\overline{x}}(y,x) \rightarrow \{0,1\}$ is well-defined iff

$$\overline{y} \in Cat_{AC} \tag{6}$$

where $Cat \in Categories$ and Cat_{AC} is the ambiguity class AC (such as `AN`, for adjective/noun ambiguity of the part of speech category) of a morphological category Cat (such as POS). For example, the function $f_{POS_{AN},A,\overline{x}}$ is well-defined ($A \in \{A,N\}$), whereas the function $f_{CASE_{145},6,\overline{x}}$ is not ($6 \notin \{1,4,5\}$). We will introduce the notation of the context part in the examples of feature value computation below. The indexes may be omitted if it is clear what category, ambiguity class, the value of the category being predicted and/or the context the feature belongs to.

The value of a well-defined feature[7] function $f_{Cat_{AC},\overline{y},\overline{x}}(y,x)$ is determined by

$$f_{Cat_{AC},\overline{y},\overline{x}}(y,x) = 1 \Leftrightarrow \overline{y} = y \wedge \overline{x} \subset x. \tag{7}$$

This definition excludes features which are positive for more than one y in any context x. This property will be used later in the feature selection algorithm.

As an example of a feature, let's assume we are predicting the category `CASE` from the ambiguity class `145`, i.e. the morphology gives us the possibility to assign nominative (`1`), accusative (`4`) or vocative (`5`) case. A feature then is e.g.

> *The resulting case is nominative* (`1`) *and the following word form is* `pracuje` *(lit. (it) works)*

[5] a subtag, i.e. (in our case) the unique value of a morphological category.

[6] Every category is, of course, treated separately. It means that e.g. the ambiguity class `23` for category `CASE` (meaning that there is an ambiguity between genitive and dative cases) is different from ambiguity class `23` for category `GRADE` or `PERSON`.

[7] From now on, we will assume that all features are well-defined.

```
AAIS1----1A--|A/A/IM/S/145/-/-/-/-/1/A/-/-/|tvrdý
NNIS1-----A--|NV/Ni/-I/S/-14/-/-/-2/-/-/A/-/-/|boj
```

Figure 2: Context where the feature $f_{POS_{NV},N,(POS_{-1}=A,CASE_{-1}=145)}$ is positive *(lit. heavy fighting).*

```
AAIS6----1A--|A/A/IMN/S/6/-/-/-/-/1/A/-/-/|pražském
NNIS6-----A--|NV/Ne/IY/S/-6/-/-/-/-/-/A/-/-/|hradě
```

Figure 3: Context where the feature $f_{POS_{NV},N,(POS_{-1}=A,CASE_{-1}=145)}$ is negative *(lit. (at the) Prague castle).*

denoted as $f_{CASE_{145},1,(FORM_{+1}=pracuje)}$, or

> *The resulting case is accusative* (4) *and the closest preceding preposition's case has the ambiguity class* 46

denoted as $f_{CASE_{145},4,(CASE_{-POS=R}=46)}$.

The feature $f_{POS_{NV},N,(POS_{-1}=A,CASE_{-1}=145)}$ will be positive in the context of Fig. 2, but not in the context of Fig. 3.

The full cross-product of all the possibilities outlined above is again restricted to those features which have actually appeared in the training data more than a certain number of times.

Using ambiguity classes instead of unique values of morphological categories for evaluating the (context part of the) features has the advantage of giving us the possibility to avoid Viterbi search during tagging. This then allows to easily add lookahead (right) context.[8]

There is no "forced relationship" among categories of the same tag. Instead, the model is allowed to learn also from the same-position "context" of the subtag being predicted. However, when *using* the model for tagging one can choose between two modes of operation: **separate**, which is the same mode used when training as described herein, and **VTC (Valid Tag Combinations)** method, which does not allow for impossible combinations of categories. See Sect. 5 for more details and for the impact on the tagging accuracy.

4 Training

4.1 Feature Weights

The usual method for computing the feature weights (the λ_i parameters) is Maximum Entropy (Berger & al., 1996). This method is generally slow, as it requires lot of computing power.

Based on our experience with tagging as well as with other projects involving statistical modeling, we assume that actually the weights are **much less important** than the features themselves.

We therefore employ very simple weight estimation. It is based on the ratio of conditional probability of y in the context defined by the feature $f_{\overline{y},\overline{x}}$ and the uniform distribution for the ambiguity class AC.

4.2 Feature Selection

The usual guiding principle for selecting features of exponential models is the Maximum Likelihood principle, i.e. the probability of the training data is being maximized. (or the cross-entropy of the model and the training data is being minimized, which is the same thing). Even though we are eventually interested in the final error rate of the resulting model, this might be the only solution in the usual source-channel setting where two independent models (a language model and a "translation" model of some sort – acoustic, real translation etc.) are being used. The improvement of one model influences the error rate of the combined model only indirectly.

This is not the case of tagging. Tagging can be seen as a "final application" problem for which we assume to have enough data at hand to train and use just one model, abandoning the source-channel paradigm. We have therefore used the **error rate** directly **as the objective function** which we try to minimize when selecting the model's features. This idea is not new, but as far as we know it has been implemented in rule-based taggers and parsers, such as (Brill, 1993a), (Brill, 1993b), (Brill, 1993c) and (Ribarov, 1996), but not in models based on probability distributions.

Let's define the set of contexts of a set of features:

$$X(F) = \{\overline{x} : \exists \overline{y}\ \exists f_{\overline{y},\overline{x}} \in F\}, \tag{8}$$

where F is some set of features of interest.

The features can therefore be grouped together based on the context they operate on. In the current implementation, we actually add features in "batches". A "batch" of features is defined as a set of features which share the same context $\overline{x}$ (see the definition below). Computationaly, adding features in batches is relatively cheap both time- and space-wise.

For example, the features

$$f_{POS_{NV},N,(POS_{-1}=A,CASE_{-1}=145)}$$

and

$$f_{POS_{NV},V,(POS_{-1}=A,CASE_{-1}=145)}$$

[8] It remains to be seen whether using the unique values – at least for the left context – and employing Viterbi would help. The results obtained so far suggest that probably not much, and if yes, then it would restrict the number of features selected rather than increase tagging accuracy.

share the context ($POS_{-1} = A, CASE_{-1} = 145$).

Let further

- F_{AC} be the pool of features available for selection.
- S_{AC} be the set of features selected so far for a model for ambiguity class AC,
- $p_{S_{AC}}(y|d)$ the probability, using model (3-5) with features S_{AC}, of subtag y in a context defined by position d in the training data, and
- $F_{AC,\overline{x}}$ be the set ("batch") of features sharing the same context $\overline{x}$, i.e.

$$F_{AC,\overline{x}} = \{f \in F_{AC} : \exists f_{\overline{y},\overline{x}} : f = f_{\overline{y},\overline{x}}\}. \quad (9)$$

Note that the size of AC is equal to the size of any batch of features ($|AC| = |F_{AC,\overline{x}}|$ for any $\overline{x}$).

The selection process then proceeds as follows:

1. For all contexts $\overline{x} \in X(F_{AC})$ do the following:
2. For all features $f = f_{\overline{y},\overline{x}} \in F_{AC,\overline{x}}$ compute their associated weights λ_f using the formula:

$$\lambda_f = \log(\frac{\tilde{p}_{AC,\overline{x}}(\overline{y})}{\frac{1}{|AC|}}), \quad (10)$$

where

$$\tilde{p}_{AC,\overline{x}}(\overline{y}) = \frac{\sum_{d=1}^{|T|} f_{\overline{y},\overline{x}}(y_d, x_d)}{\sum_{y \in AC} \sum_{d=1}^{|T|} f_{\overline{y},\overline{x}}(y, x_d)}. \quad (11)$$

3. Compute the error rate of the training data by going through it and at each position d selecting the best subtag by maximizing $p_{S_{AC} \cup F_{AC,\overline{x}}}(y|d)$ over all $y \in AC$.
4. Select such a feature set $F_{AC,\overline{x}}$ which results in the maximal improvement in the error rate of the training data and add all $f \in F_{AC,\overline{x}}$ permanently to S_{AC}; with S_{AC} now extended, start from the beginning (unless the termination condition is met),
5. Termination condition: improvement in error rate smaller than a preset minimum.

The probability defined by the formula (11) can easily be computed despite its ugly general form, as the denominator is in fact the number of (positive) occurrences of all the features from the batch defined by the context $\overline{x}$ in the training data. It also helps if the underlying ambiguity class AC is found only in a fraction of the training data, which is typically the case. Also, the size of the batch (equal to $|AC|$) is usually very small.

On top of rather roughly estimating the λ_f parameters, we use another implementation shortcut here: we do not necessarily compute the best batch of features in each iteration, but rather add all (batches of) features which improve the error rate by more than a threshold δ. This threshold is set to half the number of data items which contain the ambiguity class AC at the beginning of the loop, and then is cut in half at every iteration. The positive consequence of this shortcut (which certainly adds some unnecessary features) is that the number of iterations is much smaller than if the maximum is regularly computed at each iteration.

5 Results

We have used 130,000 words as the training set and a test set of 1000 words. There have been 378 different ambiguity classes (of subtags) across all categories.

We have used two evaluation metrics: one which evaluates each category separately and one "flat-list" error rate which is used for comparison with other methods which do not predict the morphological categories separately. We compare the new method with results obtained on Czech previously, as reported in (Hladká, 1994) and (Hajič, Hladká, 1997). The apparently high baseline when compared to previously reported experiments is undoubtedly due to the introduction of multiple models based on ambiguity classes.

In all cases, since the percentage of text tokens which are at least two-way ambiguous is about 55%, the error rate should be almost doubled if one wants to know the error rate based on ambiguous words only.

The baseline, or "smoothing-only" error rate was at 20.7 % in the test data and 22.18 % in the training data.

Table 2 presents the initial error rates for the individual categories computed using only the smoothing part of the model ($n = 0$ in equation 3).

Training took slightly under 20 hours on a Linux-powered Pentium 90, with feature adding threshold set to 4 (which means that a feature batch was not added if it improved the absolute error rate on training data by 4 errors or less). 840 (batches) of features (which corresponds to about 2000 fully specified features) have been learned. The tagging itself is (contrary to training) very fast. The average speed is about 300 words/sec. on morphologically prepared data on the same machine. The results are summarized in Table 3.

There is no apparent overtraining yet. However, it does appear when the threshold is lowered (we have tested that on a smaller set of training data consisting of 35,000 words: overtraining started to occur when the threshold was down to 2-3).

Table 4 contains comparison of the results

Category	training data	test data
POS	1.10	2.1
SUBPOS	1.06	1.1
GENDER	6.35	6.1
NUMBER	5.34	4.2
CASE	14.55	14.5
POSSGENDER	0.05	0.0
POSSNUMBER	0.13	0.1
PERSON	0.28	0.0
TENSE	0.36	0.1
GRADE	0.48	0.3
NEGATION	1.33	1.0
VOICE	0.40	0.1
VAR	0.30	0.3
Overall	22.18	20.7

Table 2: Initial Error Rate

Category	training data	test data
POS	0.02	0.9
SUBPOS	0.49	1.0
GENDER	1.78	2.0
NUMBER	2.73	0.9
CASE	6.01	5.0
POSSGENDER	0.04	0.0
POSSNUMBER	0.01	0.0
PERSON	0.12	0.0
TENSE	0.12	0.1
GRADE	0.11	0.1
NEGATION	0.25	0.0
VOICE	0.11	0.0
VAR	0.10	0.2
Overall	8.75	8.0

Table 3: Resulting Error Rate

achieved with the previous experiments on Czech tagging (Hajič, Hladká, 1997). It shows that we got more than 50% improvement on the best error rate achieved so far. Also the amount of training data used was lower than needed for the HMM experiments. We have also performed an experiment using 35,000 training words which yielded by about 4% worse results (88% combined tag accuracy).

Finally, Table 5 compares results (given different training thresholds[9]) obtained on larger training data using the "separate" prediction method discussed so far with results obtained through a modification, the key point of which is that it considers only "Valid (sub)Tag Combinations (VTC)". The probability of a tag is computed as a simple product of subtag probabilities (normalized), thus assuming subtag independence. The "winner" is presented in boldface. As expected, the overall error rate is always better using the VTC method, but some of the subtags are (sometimes) better predicted using the "separate" prediction method[10]. This could have important practical consequences – if, for example, the `POS` or `SUBPOS` is all that's interesting.

Experiment	training data size	best error rate (in %)
Unigram HMM	621,015	34.30
Rule-based (Brill's)	37,892	20.25
Trigram HMM	621,015	18.86
Bigram HMM	621,015	18.46
Exponential	35,000	12.00
Exponential	130,000	8.00
Exponential, VTC	160,000	6.20

Table 4: Comparing Various Methods

6 Conclusion and Further Research

The combined error rate results are still far below the results reported for English, but we believe that there is still room for improvement. Moreover, splitting the tags into subtags showed that "pure" part of speech (as well as the even more detailed "subpart" of speech) tagging gives actually better results than those for English.

We see several ways how to proceed to possibly improve the performance of the tagger (we are still talking here about the "single best tag" approach; the n-best case will be explored separately):

- Disambiguated tags (in the left context) plus Viterbi search. Some errors might be eliminated if features asking questions about the *disambiguated* context are being used. The disambiguated tags concentrate – or transfer – information about the more distant context. It would avoid "repeated" learning of the same or similar features for different but related disambiguation problems. The final effect on the overall accuracy is yet to be seen. Moreover, the transition function assumed by the Viterbi algorithm must be reasonably defined (approximated).
- Final re-estimation using maximum entropy. Let's imagine that after selecting all the features using the training method described here we recompute the feature weights using the usual maximum entropy objective function. This will produce better (read: more principled) weight estimates for the features already selected, but it might help as well as hurt the performance.
- Improved feature pool. This is, according to our opinion, the source of major improvement. The error analysis shows that in many cases the

[9] No overtraining occurred here either, but the results for thresholds 2-4 do not differ significantly.

[10] For English, using the Penn Treebank data, we have *always* obtained better accuracy using the VTC method (and redefinition of the tag set based on 4 categories).

Threshold:	128		16		8		4		2	
Features learned:	23		213		772		1529		4571	
Category	Sep	VTC	Sep	VTC	Sep	VTC	Sep	VTC	Sep	VTC
POS	1.50	**1.32**	0.86	**0.78**	0.66	**0.60**	0.44	**0.42**	**0.36**	0.44
SUBPOS	**1.24**	1.40	**0.78**	0.84	0.70	**0.64**	**0.36**	0.48	**0.30**	0.48
GENDER	4.50	**4.06**	3.00	**2.80**	2.40	**2.14**	2.14	**1.80**	2.08	**1.90**
NUMBER	3.46	**2.94**	2.62	**2.40**	1.86	**1.72**	1.72	**1.56**	1.80	**1.50**
CASE	11.10	**10.52**	7.74	**7.66**	**5.30**	5.34	4.82	**4.80**	4.88	**4.84**
POSSGENDER	**0.08**	0.10	**0.08**	0.12	0.08	**0.04**	**0.04**	0.06	**0.02**	0.04
POSSNUMBER	0.14	**0.04**	0.04	**0.04**	0.04	**0.00**	**0.02**	**0.02**	**0.00**	**0.00**
PERSON	0.28	**0.18**	**0.14**	0.16	0.16	**0.10**	0.14	**0.12**	0.12	**0.06**
TENSE	0.36	**0.18**	0.16	**0.14**	**0.10**	0.12	**0.10**	0.12	0.10	**0.08**
GRADE	**0.88**	1.00	0.70	**0.30**	0.44	**0.30**	0.22	**0.18**	0.22	**0.16**
NEGATION	0.62	**0.26**	**0.34**	0.36	0.28	**0.26**	**0.24**	**0.24**	0.26	**0.24**
VOICE	0.38	**0.18**	0.16	**0.14**	**0.10**	0.12	**0.10**	0.12	**0.08**	**0.08**
VAR	0.26	**0.18**	0.24	**0.22**	**0.14**	**0.14**	**0.12**	0.14	0.12	**0.04**
Overall	16.50	**13.22**	12.20	**9.58**	8.42	**6.98**	7.62	**6.22**	7.66	**6.20**

Table 5: Resulting Error Rate in % (newspaper, training size: 160,000, test size: 5000 tokens)

context to be used for disambiguation has not been used by the tagger simply because more sophisticated features have not been considered for selection. An example of such a feature, which would possibly help to solve the very hard and relatively frequent problem of disambiguating between nominative and accusative cases of certain nouns, would be a question "Is there a noun in nominative case only in the same clause?" – every clause may usually have only one noun phrase in nominative, constituting its subject. For such feature to work we will have to correctly determine or at least approximate the clause boundaries, which is obviously a nontrivial task by itself.

7 Acknowledgements

Various parts of this work has been supported by the following grants: Open Foundation RSS/HESP 195/1995, Grant Agency of the Czech Republic (GAČR) 405/96/K214, and Ministry of Education Project No. VS96151. The authors would also like to thank Fred Jelinek of CLSP JHU Baltimore for valuable comments and suggestions which helped to improve this paper a lot.

References

Adam Berger, Stephen Della Pietra, Vincent Della Pietra. 1996. Maximum Entropy Approach. In *Computational Linguistics*, vol. 3, MIT Press, Cambridge, MA.

Eric Brill. 1993a. A Corpus Based Approach To Language Learning. *PhD Dissertation*, Department of Computer and Information Science, University of Pennsylvania.

Eric Brill. 1993b. Automatic grammar induction and parsing free text: A Transformation-Based Approach. In: *Proceedings of the 3rd International Workshop on Parsing Tech nologies*, Tilburg, The Netherlands.

Eric Brill. 1993c. Transformation-Based Error-Driven Parsing. In: *Proceedings of the Twelfth National Conference on Artificial Intelligence.*

Kenneth W. Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143, Austin, Texas. Association for Computational Linguistics, Morristown, New Jersey.

Jan Hajič. 1994. Unification Morphology Grammar. PhD Dissertation. MFF UK, Charles University, Prague.

Jan Hajič. In prep. Automatic Processing of Czech: between Morphology and Syntax. MFF UK, Charles University, Prague.

Jan Hajič, Barbora Hladká. 1997. Tagging of Inflective Languages: a Comparison. In *Proceedings of the ANLP'97*, pages 136–143, Washington, DC. Association for Computational Linguistics, Morristown, New Jersey.

Barbora Hladká. 1994. Programové vybavení pro zpracování velkých českých textových korpusů. *MSc Thesis*, Institute of Formal and Applied Linguistics, Charles University, Prague, Czech Republic.

Bernard Merialdo. 1992. Tagging Text With A Probabilistic Model. Computational Linguistics, 20(2):155–171

Kiril Ribarov. 1996. Automatická tvorba gramatiky přirozeného jazyka. *MSc Thesis*, Institute of Formal and Applied Linguistics, Charles University, Prague, Czech Republic. In Czech.