# Validating the Quality of Full Morphological Annotation

**Drahomíra „johanka“ Spoustová, Pavel Pecina, Jan Hajič, Miroslav Spousta**

Institute of Formal and Applied Linguistics
Charles University Prague, Czech Republic
{johanka,pecina,hajic,spousta}@ufal.mff.cuni.cz

**Abstract**

In our paper we present a methodology used for low-cost validation of quality of Part-of-Speech annotation of the *Prague Dependency Treebank* based on multiple re-annotation of data samples carefully selected with the help of several different Part-of-Speech taggers.

## 1. Introduction

All supervised machine-learning methods rely on quality and extent of training data – in the area of Computation Linguistics and Natural Language Processing often in a form of a manually annotated corpus. While the amount of data is usually only an issue of time and costs, the quality assurance is a non-trivial process. Each annotated corpus should be provided with information regarding its annotation quality, such as inter-annotator agreement or kappa measure (cf. (Artstein and Poesio, 2007)). Absence of such a measure leads to insufficient performance evaluation of employed machine-learning methods – performance of any method should be never reported without mentioning how difficult the task is, for example for a human.

Quality of corpus annotation is affected mainly by following two factors: annotator's errors and imperfect specification of annotation guidelines. The errors can be detected by multiple annotation of the same data and combining the results e.g. by voting – it is unlikely that more annotators make the same mistake at the same place. The latter, however, is a real problem. Too vague or too detailed specification of the annotation guidelines can lead to a situation when different decisions of multiple annotators are equally correct. Frequency estimation of these two cases in a corpus is essential for estimating possible room for improvement of employed methods.

## 2. Czech Part-of-Speech tagging

Part-of-Speech tagging is a process of assigning particular part-of-speech tag to the words in a text. Czech as a language with very rich morphology distinguishes up to 12 different morphological categories for each word (Part of speech, Detailed part of speech, Gender, Number, Case, Possessor's gender, Possessor's number, Person, Tense, Degree of comparison, Negation, and Voice) creating quite detailed and complex morphological tags with more than 4200 possible values (the size of tagset) (Hajič, 2004).

The process of tagging consists of three steps: 1) during *morphological analysis* each word form in a text is assigned a list of all possible tags from a morphological dictionary, 2) if the word form does not appear in the dictionary the *guesser* attempts to assign ("guess") possible tags based on the word ending, 3) for each word form the *tagger* selects the most likely tag from the list. Evaluating tagger means, in fact, evaluating quality of all the steps including the dictionary, guesser, and tagger.

The Prague Dependency Treebank version 2 (PDT2) contains a large amount of Czech texts with morphological, syntactic, and semantic annotation (Hajič et al., 2006). Almost two million words annotated on the morphological level is split into three parts: *train* for training purposes, *dtest* for development purposes, and *etest* for evaluation (Table 1 shows exact number of tokens). The morphological annotation of PDT version 1 was originally performed by multiple annotators and subsequently combined into one reference annotation (Hajič et al., 2006). Some additional semi-manual corrections were also performed prior the release of version 2 by several other annotators (Štěpánek, 2006). As a consequence of this process, no exact details of the annotation quality are known.

| *data set* | *size* |
|---|---|
| train | 1,539,241 |
| dtest | 201,651 |
| etest | 219,765 |

Table 1: PDT 2.0 size (tokens)

Current Czech state-of-the-art taggers developed and trained on the Prague Dependency Treebank version 2 include the Feature-based tagger by Hajič (Hajič, 2004), HMM-based tagger by Krbec (Krbec, 2005), and Morče tagger (Votrubec, 2006) based on averaged perceptron. They achieve accuracy around 95% (details shown in Table 2) and practically no significant gain in performance was achieved in last years. This poses a question whether the taggers already reached their limits and the data quality prevents their further improvement or not.

| *Tagger* | *train* | *dtest* | *etest* |
|---|---|---|---|
| Feature-based | 96.36 % | 94.28 % | 94.04 % |
| HMM | 98.78 % | 95.13 % | 94.82 % |
| Morče | 97.70 % | 95.43 % | 95.12 % |

Table 2: Accuracy of current state-of-the-art Czech taggers

### 2.1. The HMM tagger

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

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

where

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

The trigram probability $P(W \mid T)$ in formula 2 replaces 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).
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:

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

where $w(C,T)$ is the transition weight for tag $T$ in context $C$, $n$ is number of features, $\alpha_i$ 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 ($\alpha$) 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.

### 2.3. 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:

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

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

## 3. Methodology Overview

A quite straightforward way how to validate quality of a corpus annotation and find annotation errors is an independent re-annotation of the entire data (a new annotator, the same guidelines). Performing more than one re-annotation brings in the possibility of combining the results to detect potential errors of new annotators and avoid them by voting.
Parallel annotation of the whole data is obviously a time consuming and very expensive process. We can, of course, reduce the cost of the new annotation by processing only a random sample of the data and use the inter-annotator agreement on this sample as an estimation of the quality of the whole corpus. However, this method has two disadvantages. First, it detects only a limited subset of incorrectly annotated words – the smaller the sample, the less incorrectly annotated words we detect (and eventually can correct). Second, it does not distinguish between disagreement caused by the annotator's keying mistakes or by imperfect annotation guidelines.
To solve the task of validating and correcting the annotation, keeping minimal costs and avoiding the problems mentioned above, we propose the following procedure how to carefully select the data for re-annotation.
First, we identify all *trivial data* – the tokens with only one possible tag that can be excluded from any further analysis. Second, we apply several (at least three) state-of-the-art automatic systems (taggers) based on different methods on the entire corpus (except the evaluation data of course) and select the tokens that were assigned a correct tag congruently

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

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

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

by all systems (all taggers agreed on the reference tag). We call this selection *easy data* and assume that these cases are annotated correctly – they do not mean any problem for the current systems and are solved easily. The remaining subset comprises all *problematic data* – these are the positions causing troubles for at least one system, which we interpret as a sign of their eventual incorrect annotation and a reason for the in depth analysis.

The subset of corpus data for the re-annotation will consist only of shuffled random sample of the *easy* and *problematic data*. A smaller sample of the *easy data* will be used to assure the quality of annotators' work (we can expect 100% agreement between annotators themselves as well as between annotators and the reference data). The main annotators' effort will be focused on a larger sample of the *problematic data* in order to estimate the inter-annotator agreement and analyze potential room for systems' performance improvement.

## 4. Application on PDT

As a first step, morphological analysis and guesser (version from April 2006) were applied on the *train* set and *dtest* set of PDT2. Thus we identified 44.12% and 43.11%, respectively, of the tokens as *trivial*. Then the three taggers mentioned in Section 2 (trained on the PDT2 *train* set) applied on the same data sets agreed on 51.23% and 47.41%, respectively, cases to be the *easy data*. Finally, the *problematic data* comprised 4.65% of the *train set* and 9.48% of the *dtest* set (see details in Table 3).

| *data* | *train* | | *dtest* | |
|---|---|---|---|---|
| trivial | 679,061 | 44.12 % | 86,922 | 43.11 % |
| easy | 788,573 | 51.23 % | 95,604 | 47.41 % |
| problematic | 71,607 | 4.65 % | 19,125 | 9.48 % |
| total | 1,539,241 | 100 % | 201,651 | 100 % |

Table 3: Size of particular parts of the data

Since our main interest lies in analysis of the *problematic data* we selected a quite large sample of them from the *dtest* set (25%, 5,000 tokens) and only half the size from the *easy data* (2,500 tokens). The same amounts of tokens were sampled and added also from the *train* set and the entire set of 15,000 tokens was independently annotated by three human annotators at an average speed of 1000 tokens per day. The tokens to be annotated were randomly shuffled and presented to the annotators independently from each other with a context of one preceding, the current, and one following sentence. A list of possible tags for each word was obtained from the morphological analyzer and the guesser (version also from April 2006) and enriched by the tag from the reference annotation in case it did not appear in the morphological dictionary and was not proposed even by the guesser.

## 5. Results

### 5.1. Inter-annotator Agreement

Inter-annotator agreement is the basic measure of corpus annotation quality and difficultness of a task. We measured its value separately on the *easy* and *problematic* data on both the *dtest* and *train* sample. All results are presented in Table 4 and we can conclude that the work of annotators was quite reliable. The best annotator (A2) achieved 99.04% agreement with the reference PDT annotation on the *dtest easy* sample and 98.52% agreement on the *train easy* data sample.

### 5.2. Detailed Analysis

The data we obtained from the three annotations and the agreement results allow us to make more thorough analysis of the quality of the whole corpus. We can distinguish the following three cases for each annotated token:

**Correct annotation** At least two annotators agree with the reference annotation (this eliminates an eventual error of one annotator).

**Incorrect annotation** All three annotators agree with each other and the reference annotation differs from their choice (a sign that the reference tag is probably wrong).

**Vague annotation** All other cases (multiple tags are equally correct or errors by multiple annotators). We are interested only in the first case, but we can not distinguish it from the case of multiple errors, so we have only the upper limit for the amount of really vague tags.

| | *data* | *all* | *corr.* | *incorr.* | *vague* |
|---|---|---|---|---|---|
| *dtest* | easy | 2,500 | 2,482 | 4 | 14 |
| | problematic | 5,000 | 4,605 | 171 | 224 |
| *train* | easy | 2,500 | 2,471 | 13 | 15 |
| | problematic | 5,000 | 4,458 | 255 | 287 |

Table 5: Correctness of the annotated data (number of tokens)

Counts of these cases in annotated data are presented in Table 5. From this evidence we can estimate their distribution on the entire PDT test sets which is shown in Table 6. 98.99 % of tokens in *dtest* set are annotated correctly, 0.37 % of tokens are very likely to be annotated incorrectly, and annotation of (up to) 0.65 % of *dtest* tokens is vague. Similar results were estimated for the *train* set.

| | *data* | *size* | *corr.* | *incorr.* | *vague* |
|---|---|---|---|---|---|
| *dtest* | easy | 95,604 | 99.28 | 0.16 | 0.56 |
| | problematic | 19,125 | 92.10 | 3.42 | 4.48 |
| | all, weighted | 201,651 | **98.99** | **0.37** | **0.65** |
| *train* | easy | 788,573 | 98.84 | 0.52 | 0.64 |
| | problematic | 71,607 | 89.16 | 5.10 | 5.74 |
| | all, weighted | 1,539,241 | **98.90** | **0.50** | **0.59** |

Table 6: Correctness estimation for the whole *dtest* and *train* set

Finally, we extend this estimation to the entire PDT (including *etest*) and conclude that

| | *data* | *size* | *A1* | *A2* | *A3* | *voted* |
|---|---|---|---|---|---|---|
| *dtest* | easy | 2,500 | 97.00 | 99.04 | 98.36 | 99.32 |
| | problematic | 5,000 | 88.48 | 92.86 | 88.46 | 92.48 |
| *train* | easy | 2,500 | 97.64 | 98.52 | 97.92 | 98.92 |
| | problematic | 5,000 | 86.66 | 90.12 | 81.46 | 89.88 |

Table 4: Annotator (A1–A3) agreement with the reference annotation measured on the annotated data samples (in %).

- 1,939,314 tokens (98.91 %) are annotated with correct tags and are a reliable source of linguistic evidence
- 9,563 tokens (0.49 %) are annotated with incorrect tags and should be identified and corrected
- (up to) 11,780 tokens (0.60 %) are vague tags (undecidable ambiguities, foreign words etc.). Detection and linguistic analysis of these tags should lead to adjustment of the annotation guidelines.

## 6. Conclusion

We have proposed a method how to validate the corpus annotation quality and detect large subset of particular problematic and vague tags with minimal costs. Our method can be used for any language and a wide range of annotation, the only necessity is a set of automatic methods based on different principles.

We have shown that even tagging of the training data can be useful. For example, if we randomly choose for re-annotation another 10 000 non-*trivial* tokens from the training data, we can expect to find about 90 annotation errors and 103 vague tags, compared to 510 errors and 574 vague tags using our method of data selection.

Obviously, we can not find all the problematic tags using this method, but we can effectively detect more than one half of them by re-annotating only one tenth of the non-*trivial* data (instead of one half, when selecting the data randomly).

The second result of our experiment is the correctness estimation for the whole PDT. We can conclude that the morphological annotation has a very high quality and the taggers have still some room for their improvement without correcting the data. The amount of vague tags is also reasonable.

## Acknowledgements

The research described here was supported by the projects *MSM0021620838* and *LC536* of *Ministry of Education, Youth and Sports* of the Czech Republic, *GD201/05/H014* of the *Grant Agency* of the Czech Republic and *1ET201120505* of the *Grant Agency of the Academy of Sciences* of the Czech Republic.

## 7. References

Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. Submitted to Computational Linguistics. (2007)

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

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

Jan Hajič. Disambiguation of Rich Inflection (Computational Morphology of Czech). Vol. 1. Charles University Press Prague. (2004)

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

Jan Štěpánek. Post-annotation Checking of Prague Dependency Treebank 2.0 Data *In: Proceedings of the 9th International Conference*, TSD 2006, Springer-Verlag Berlin Heidelberg. pp. 277-284. (2006)

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

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