

# Trigram morphosyntactic tagger for Polish

Łukasz Dębowski<sup>1</sup>

Polish Academy of Sciences, Institute of Computer Science, ul. Ordona 21, 01-237  
Warszawa, Poland

**Abstract.** We introduce an implementation of a plain trigram part-of-speech tagger which appears to work well on Polish texts. At this moment the tagger achieves 9.4% error rate, which makes it significantly better than our previous stochastic disambiguator. Since the trigram model for Polish behaves similarly to Czech, we hope to reach Czech state-of-art error rate when the quality of the training data improves.

## 1 Introduction

In a recent paper [2], we proposed a scheme of a stochastic morphosyntactic tagger (contextual disambiguator of morphological analyser) for inflective languages with large and highly structured tagset (such as Polish, Czech, or other Slavic languages). The approach assumed that tags are lists of discrete attributes, the probability distribution for each attribute is modeled separately, and the probability of the whole tag is constructed from the probability of its attributes according to Naive Bayes assumption. The resulted implementation of the tagger happened to be very slow and prone to “underlearning”: The tagger trained on 10k word data performed as well as that trained on 500k word data, and its error rate of 20% seemed impossible to reduce. On the other hand, it is known that smoothed trigram-based taggers achieve about 10% error rate for Czech [4], while some simple extension of the model, described in [7], can commit as little as 5% errors [3]. Therefore we decided to implement a new,  $n$ -gram-based tagger (own implementation is the best way to experiment with underspecified details of the general framework). Our results for Polish are comparable to Czech: unigram tagger commits 19%, bigram tagger — 13%, and smoothed trigram tagger — 9.4% errors. We hope that some further reduction of the error rate is possible by the improvement of the training data themselves. (The manual annotation has not been completed yet and *Morfeusz*, the morphological analyser which has been being developed at our institute and which we use, still makes some systematic errors.) If the improvement occurs, the tagger will be used for automatic annotation of a 100M word corpus of Polish [1].

## 2 The model

The task of morphosyntactic disambiguation consists in choosing a string of contextually suitable tags out of a string of tag alternatives given for a ran-

dom string of text words by a morphological analyser. Let us introduce the variables:  $W_i$  – word-form at  $i$ -th text position,  $\mathcal{T}_i = M(W_i)$  – morphological analysis of  $W_i$  (a set of morphosyntactic tags),  $T_i \in \mathcal{T}_i$  – the contextually valid tag. Contextually valid tags  $T_{1:n} = (T_1, T_2, \dots, T_n)$  can be determined by humans for sufficiently long  $W_{1:n} = (W_1, W_2, \dots, W_n)$  (almost) uniquely but the dependence between these two strings is very complex. The automatic computation of  $T_1, T_2, \dots, T_n$  can be made efficient if some error rate is allowed (even humans disagree for about 3% of tokens in the annotation scheme proposed in [6]). One of heuristic approaches which appears unexpectedly fruitful is trigram model and its modifications. In this model, the dependence between  $T_1, T_2, \dots, T_n$  and  $W_1, W_2, \dots, W_n$  is modeled by a stationary Bayesian network depicted in Fig. 1. It is assumed that the correct tags  $T_i$  for words  $W_i$  are given by the maximum of conditional probability

$$P(T_{1:n}|W_{1:n}) = \max. \quad (1)$$

(We hide the distinction between random variables and their values as well as we reverse the usual order of string to make the next formulae shorter.)

The only difficulty that remains concerns estimating probabilities  $P(T_0|T_{1:2})$  and  $P(W_1|T_{1:2})$ . The number of word types is potentially infinite and the number of tag types is of order 1k–10k so hardly all tag types appear in the usual training data (500k–1M words). To make  $P(U_0|T_{1:2})$  with  $U_0 \equiv T_0$  or  $U_0 \equiv W_1$  both determinate and non-zero, we cannot estimate them as  $P(U_0|T_{1:2}) = n(U_0, T_{1:2})/n(T_{1:2})$ , where  $n(\cdot)$  stands for the count of particular event in the training data. Instead of this, we apply both linear interpolation and back-off smoothing. For  $m \geq 1$  we put

$$P(U_0|T_{1:2}) = P_L(U_0|T_{1:2}), \quad (2)$$

$$P_L(U_0|T_{1:m}) = \lambda_m P_B(U_0|T_{1:m}) + (1 - \lambda_m) P_L(U_0|T_{1:m-1}), \quad (3)$$

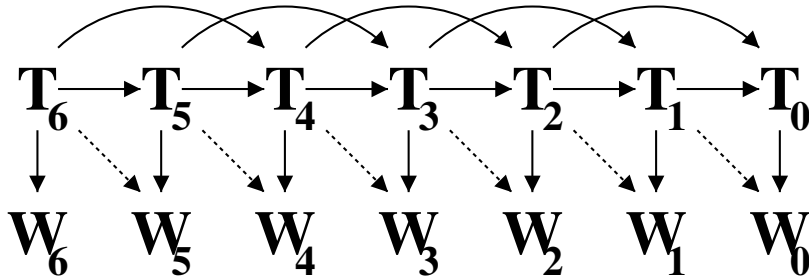
$$P_B(U_0|T_{1:m}) = \begin{cases} \frac{n(U_0, T_{1:m})}{n(T_{1:m}) + c_m} & \text{if } n(U_0, T_{1:m}) > 0, \\ \frac{c_m}{n(T_{1:m}) + c_m} \cdot \frac{P_B(U_0|T_{1:m-1})}{D(T_{1:m})} & \text{else,} \end{cases} \quad (4)$$

$$D(T_{1:m}) = \sum_{U_0: n(U_0, T_{1:m})=0} P_B(U_0|T_{1:m-1}), \quad (5)$$

where  $T_{1:0} \equiv \emptyset$ . The iterations begin with

$$P_L(U_0|\emptyset) = P_B(U_0|\emptyset) = \begin{cases} \frac{n(U_0)}{N+c_0} & \text{if } n(U_0) > 0, \\ \frac{c_0}{N+c_0} \cdot \frac{1}{L} & \text{else,} \end{cases} \quad (6)$$

where  $N$  is the size of the training data. Constant  $L$  is chosen as  $L = 1$  for  $U_0 \equiv W_1$  while for  $U_0 \equiv T_0$ ,  $L$  is the number of these possible values of  $T_0$  which are given by the morphological analysis but were not seen in the training data. Parameters  $0 \geq \lambda_1 \geq 1$ ,  $0 \geq \lambda_2 \geq 1$ ,  $c_0 > 0$ ,  $c_1 > 0$ ,  $c_2 > 0$  are chosen empirically so as to minimize the error rate directly. In particular,



**Fig. 1.** Bayesian networks used for tagging. (For the graph convention, see explanation in [5].) Solid arrows correspond to the standard trigram model. Dotted arrows are added in the modification of [7]

parameters  $\lambda_m$  and  $c_m$  can obtain very different values for  $U_0 \equiv T_0$  ( $\lambda_m^T$  and  $c_m^T$ ) and for  $U_0 \equiv W_1$  ( $\lambda_m^W$  and  $c_m^W$ ).

Yet another approximation we use consists in replacing rare literal word-forms by strings representing their morphological analyses. If  $\tilde{W}_i$  is the literal word-form encountered in the text and  $M(\tilde{W}_i)$  is its morphological analysis, we determine the value of  $W_i$  in the following way,

$$W_i = \begin{cases} \tilde{W}_i & \text{if } n(\tilde{W}_i) \geq c_M, \\ M(\tilde{W}_i) & \text{else,} \end{cases} \quad (7)$$

where  $c_M$  is another parameter to be optimized.

### 3 Preliminary results

At the moment, we have not implemented any algorithm for the optimization of parameters  $\lambda_m$  and  $c_m$ . The error rate, however, seems to depend on them very smoothly, and it seems that the global minimum can be reached by manual coordinate-wise test-and-trial. In table 1 we collect several results. All error rates were obtained for 590k token training corpus and 4200 token test corpus. The set of annotated tokens includes both words and punctuation. The test corpus was used as a smoothing corpus to speed up evaluation. This can fake the estimates of the true error rates but it is worth noting that the final improvement of the tagger (downto 9.4% error rate) was resulted by a slight revision of the training data for the fixed old parameters: 1) some systematic mistakes of the morphological analyser were corrected, 2) a procedure for the correction of typos in the training data was revised as well. It is worth noting that both changes slightly diminished both the ambiguity rates and the error rates but the error rate divided by the ambiguity rate dropped. (This concerned especially the gender.)

The high optimal value of  $c_1^W$  is unexpected. It is roughly equal to the size of the training data lexicon, so it may depend on the size of the training

**Table 1.** Error rate against the model parameters

order	$\lambda_1^T$	$\lambda_2^T$	$c_i^T$	$\lambda_1^W$	$\lambda_2^W$	$c_1^W$	$c_2^W$	$c_M$	search states	error rate
unigram	0	–	1	1	0	1	–	1	20	18.8%
bigram	1	0	1	1	0	1	–	1	20	13.0%
trigram	1	1	1	1	0	1	–	1	20	12.7%
trigram	1	0.3	1	1	0	1	–	1	20	12.4%
trigram	1	0.3	0.1	1	0	1	–	1	20	12.2%
trigram	1	0.3	0.1	1	0	30k	–	1	20	11.1%
trigram	1	0.3	0.1	1	0	30k	–	1	50	10.3%
trigram	1	0.5	0.1	1	0	30k	–	1	50	10.1%
trigram	1	0.5	0.1	1	0.27	30k	1	1	50	9.9%
trigram	1	0.5	0.1	1	0.27	30k	1	3	50	9.6%
trigram <sup>a</sup>	1	0.5	0.1	1	0.27	30k	1	3	50	9.4%

<sup>a</sup> After some improvement of the training data.

data. On the other hand,  $c_2^W = 1$  is locally optimal. The parameter called “search states” results from replacing the exact Viterbi search to compute maximum (1) by its approximation called beam search, where the number of search states per token is bounded artificially. Although the number of search states can reach  $\gg 20^3$  for many a sequence of three Polish adjectives, increasing the number of beam search states beyond 50 does not contribute to a significant reduction of the error rate.

For a linguistic discussion, it is interesting to remind that Polish morphosyntactic tags are vectors of qualitatively different attributes. In table 2 we present the ambiguity rate of some principal attributes and the error rate against words with the ambiguity of a given attribute. According to table 2, the part-of-speech and the gender is much harder to determine for a word with a part-of-speech ambiguity than is the case for a word with a case ambiguity! Does it contradict a widely shared belief in the difficulty of disambiguating Slavic nominative-accusative syncretism (our third cause of errors)?

Some explanation of the high part-of-speech error rate may be a systemic artificial homonymy that is assumed in the training data. Many Polish wordforms historically derived as gerunds or participles, such as “oświecenie”, possess two meanings: one closer to a noun/adjective (“oświecenie” = enlightenment) and one closer to a verb (“oświecenie” = illumination). In the adopted annotation scheme [6], such a distinction exists for any Polish gerund or participle even if one can hardly figure out the noun-like meaning. Both human anotators and the tagger get confused, which is reflected in the

**Table 2.** Ambiguity rate and error rate against tag attribute

attribute	ratio of tokens with ambiguous attribute	attribute error rate against all tokens	attribute error rate against tokens with ambiguous attribute
any	55.5%	9.4%	17.6%
POS	13.3%	2.1%	16.0%
number	18.2%	1.3%	7.1%
case	49.1%	4.7%	9.5%
gender	31.4%	4.6%	14.5%
person	3.4%	0.2%	6.1%
degree	4.4%	0.4%	9.0%
aspect	6.8%	1.3%	18.3%
negation	3.8%	1.1%	28.0%

high error rate on the attributes of aspect and negation (definite for verbal and indefinite for non-verbal forms). Similarly, the high error rate on gender might be resulted by the gender classification including as much as 9 different values.

## References

1. P. Bański, A. Przepiórkowski, A. Kupść, Ł. Dębowski, M. Marciniak, and A. Mykowiecka. The design of the IPI PAN corpus. In *PALC 2001: Practical Applications in Language Corpora*, pages 225–232. Peter Lang – Europäischer Verlag der Wissenschaften, 2003.
2. Łukasz Dębowski. A reconfigurable stochastic tagger for languages with complex tag structure. In *Proceedings of Morphological Processing of Slavic Languages. EACL'03, Budapest*, pages 63–70. 2003.
3. Jan Hajič, Pavel Krbec, Pavel Květoň, Karel Oliva, and Vladimír Petkevič. Serial combination of rules and statistics: A case study in Czech tagging. In *Proceedings of ACL'01, Toulouse*, pages 260–267. 2001.
4. Jiří Mirovský. Morfologické značkování textu: automatická disambiguace. Master thesis. UFAL, Faculty of Mathematics and Physics, Charles University, Prague.
5. Kevin Murphy. A brief introduction to graphical models and Bayesian networks. <http://www.ai.mit.edu/~murphyk/Bayes/bayes.html>.
6. Adam Przepiórkowski and Marcin Woliński. A flexemic tagset for Polish. In *Proceedings of Morphological Processing of Slavic Languages, EACL 2003*, pages 33–40, 2003.
7. Scott M. Thede and Mary P. Harper. A second-order hidden Markov model for part-of-speech tagging. In *Proceedings of ACL'99, College Park*, pages 175–182. 1999.