Improving Data Driven Part-of-Speech Tagging by Morphologic Knowledge Induction

Uwe D. Reichel

Department of Phonetics and Speech Communication
University of Munich, Schellingstr. 3, 80799 Munich, Germany
reichelu@phonetik.uni-muenchen.de

Abstract. We present a Markov part-of-speech tagger for which the $P(w|t)$ emission probabilities of word w given tag t are replaced by a linear interpolation of tag emission probabilities given a list of representations of w. As word representations, string suffixes of w are cut off at the local maxima of the Normalized Backward Successor Variety. This procedure allows for the derivation of linguistically meaningful string suffixes that may relate to certain POS labels. Since no linguistic knowledge is needed, the procedure is language independent. Basic Markov model part-of-speech taggers are significantly outperformed by our model.

1 Introduction

There are two main reasons why part-of-speech (POS) tagging cannot simply be carried out by lexicon lookup:

- The same word can be related to different POS labels depending on its context.
- During application the tagger most certainly will be confronted with words not given in the lexicon (*out-of-vocabulary, OOV* cases).

To face these problems the following methods have already been established:

- Take some contextual information into account.
- Examine also substrings of unknown words that have been seen in the training data with a higher probability.

Contextual information can consist of the word environment (as in [12]) and the preceding POS tags (as in all data-driven Markov taggers). For inflective languages where suffixes bear POS information, the extracted substrings generally consist of string suffixes of several fixed lengths (c.f. [1], [10]).

In general, the available POS taggers can be divided into rule-based and data-driven ones. Rule-based approaches like ENGTWOL [12] operate on a) dictionaries containing word forms together with the associated POS labels and morphologic and syntactical features like sub-categorization frames and b) context sensitive rules to choose the appropriate labels during application.

Their major drawbacks are:

- time consuming rule adjustment,
- lack of generalization capability, and
- missing transferability to other languages.

Among data-driven approaches there are Markov taggers [4], Maximum Entropy-based taggers [6], and hybrid models like the Tree Tagger [9] combining the Markov framework and a decision tree classifier, and finally Transformation-based taggers [2]. In Markov systems, generally the most probable tag sequence given the observed word sequence is estimated (see Section 3). Maximum entropy approaches are able to integrate influences on tagging from a variety of information sources. However, also Markov approaches, as the one in this study, can be satisfyingly enriched by additional influence factors. Transformation-based tagging finally is a synergy of statistical and rule-based approaches; it derives tagging disambiguation rules by statistical means from a set of rule templates.

2 Goal of this paper

As already mentioned, a possible way to cope with OOV cases is to find a linguistically meaningful word representation that has been observed in the training data with a high probability. For a language-independent data-driven approach it is highly eligible to derive such representations solely by statistical means.

All approaches listed above show shortcomings in one or both of these respects. Transformation-based taggers do not give specifications for this task, and adding rule templates for linguistically motivated word representations anyway would lead to a large increase of calculation costs accompanying the enlargement of the template set. The Tree Tagger approach [9] does not contain an automatic retrieval of word representations, since the tagger must be provided with a language dependent suffix inventory. The TnT-tagger [1] and Synther [10] use string suffixes of some fixed lengths that are not appropriately related to linguistic entities, since these approaches do not sufficiently take into account the variable size of linguistically meaningful suffixes.

In this study we attempt to enhance POS tagging by providing it with linguistically more meaningful entities derived in a purely data-driven manner.

In the following, classical Markov Part-of-Speech Tagging will be revised and our extensions to this basic model will be presented. Further it will be shown, how linguistically relevant word representations can be obtained in the form of string suffixes of flexible length.

3 Markov Part-of-Speech Tagging

3.1 Basic Form of a Markov POS Tagger

The aim as formulated in [4] is to estimate the most probable tag sequence $\hat{T}$ given word sequence W:

$$\hat{T} = \arg\max_{T} \Big[P(T|W) \Big]. \tag{1}$$

To estimate $P(T|W)$, first, a reformulation is needed by applying the Bayes Formula, which leads to:

$$\hat{T} = \arg\max_{T} \Big[P(T)P(W|T) \Big], \tag{2}$$

given that the denominator $P(W)$ is constant. Further, two simplifying assumptions are to be made to get reliable counts for the probability estimations:

- The probability of word w_i depends only on its tag t_i
- The probability of tag t_i depends only on a limited tag history $t\text{-}history_i$

The resulting formula is thus:

$$\hat{T} = \arg\max_{t_1 \ldots t_n} \Big[\prod_{i=1}^{n} P(t_i|\text{t-history}_i) P(w_i|t_i) \Big]. \tag{3}$$

$\hat{T}$ is retrieved using the Viterbi algorithm [11].

4 Generalizations of the basic model

In equation [3] first $P(t_i|\text{t-history}_i)$ is replaced by a linearly interpolated trigram model

$$\sum_j u_j P(t_i|\text{t-history}_{ij}),$$

j ranging from unigram to trigram tag history. Further, w_i is replaced by a list of word representations leading to a reformulation of $P(w_i|t_i)$ by

$$\frac{P(w_i)}{P(t_i)} \sum_k v_k P(t_i|\text{w-representation}_{ik}),$$

again applying Bayes formula and linear interpolation. Our model is thus given by:

$$\hat{T} = \arg\max_{t_1 \ldots t_n} \Big[\prod_{i=1}^{n} \frac{1}{P(t_i)} \sum_j u_j P(\text{t}_i|\text{t-history}_{ij}) \sum_k v_k P(\text{t}_i|\text{w-representation}_{ik}) \Big] \tag{4}$$

omitting again constant $P(W)$. The interpolation weights u_j and v_k in equation [4] are calculated via the EM algorithm [3]. The probability distributions are smoothed by absolute discounting.

In order to reduce calculation effort in application, solely for unknown words the probabilities are calculated for all POS tags. For known words, only the POS tags co-occurring with them in the training corpus are taken into consideration.

5 Word representations

The representation of words seen in the training data is simply the word form. For OOV cases the representation is given by string suffixes which are determined by Normalized Backward Successor Variety (NBSV). The Successor Variety (SV) of a string is defined as the number of different characters that follow it in a given lexicon. This concept is adopted from stemming procedures like the Peak and Plateau algorithm of [5]. Backward SV means that the SVs are calculated from reversed strings in order to increase the probability to separate linguistically meaningful *suffixes*. In our approach the SVs are weighted with respect to the mean SV at the corresponding string position to eliminate positional effects. The mean SV is highest in the beginning and declines continuously while moving forward in the word string as can be seen in Figure 1.

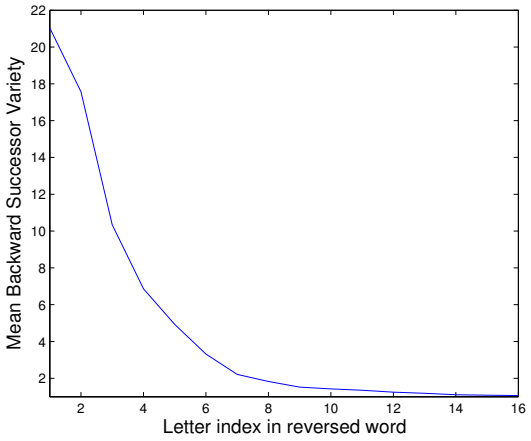

Fig. 1. Mean Backward Successor Variety declining in dependence of the in-word position.

The lexicon of reversed words is represented in the form of a trie (cf. Figure 2), in which the SV at a given state is the number of all outgoing transitions. NBSV peaks are treated as morpheme boundaries. Since this method is knowledge-free, of course not all of the obtained segments necessarily correspond to linguistic meaningful entities as might be suggested by Figure 2.

The two suffixes derived by the first two NBSV maxima are used as word representations. An example is given in Table 1. Here for *Aktivierungen (activations)* the word representations *en, ungen* (inflectional plural ending and a derivational noun suffix + ending respectively) are derived.

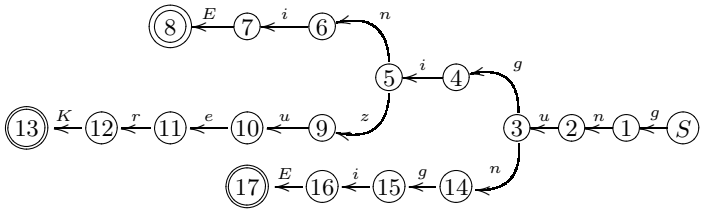

Fig. 2. Lexicon trie reversely storing the entries *Einigung (agreement), Kreuzigung (crucifixion)* and *Eignung (adequacy)*. The SV peaks at nodes 3 and 5 correspond to the boundaries of the morphemes *ung* (noun suffix) and *ig* (adjective suffix), respectively.

Table 1. Derivation of word representations for the word *Aktivierungen* (*activations*) from Normalized Backward Successor Variety (NBSV) peaks.

reversal: Aktivierungen ⟶ negnureivitkA								
letters	n	**e**	g	n	**u**	r	e	...
NBSV	0.9	**1.5**	1.3	0.7	**4.3**	2.4	5.0	...
word representations: **en, ungen**								

6 Data

The data comprised 382400 tokens taken from German connected text mostly from the "European Corpus Initiative Multilingual Corpus 1" CD-ROM, pre-tagged by the IMS Tree Tagger [9] and partially hand corrected. 85% were used for training, and 15% as test set. The percentage of OOVs in the test data was 38.12% for word types and 11.52% for word tokens. The Stuttgart Tübingen tagset [8] was used which comprised 54 tags for our data.

7 Results

In Table 2 the results are given in the form of tagging accuracy as well as the κ score and the Conditional Relative Entropy (CRE) of the reference test data and the tagger output.

κ score. κ is defined the following way:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}, \tag{5}$$

$P(A)$ being the proportion of correctly classified words and $P(E)$ the expected proportion of words correctly classified by chance. Thus, κ corrects accuracy with respect to the tagset size, accounting for the fact that the tagging task becomes more difficult with increasing size.

Conditional relative entropy. In order to compare the syntactic divergence between the reference data and the tagger output, we employed the CRE measure, which we had already used to quantify the adjustment of reference phonotactics in grapheme-to-phoneme conversion systems in [7]. For syntactic comparison by means of CRE, syntax has been directly and crudely represented by the POS sequences. CRE is given by the following equation:

$$\begin{aligned} \mathrm{CRE} &= D(p(y|x) \| q(y|x)) \\ &= \sum_x p(x) \sum_y p(y|x) \log \frac{p(y|x)}{q(y|x)}. \end{aligned} \tag{6}$$

In equation [6] p and q are the conditional probabilities of the POS tag y given the tag history x. p is derived from the original data and q from the output of our POS tagger. The relative entropy $D(p\|q)$ is a measure of the divergence of the probability distributions p and q expressed in the average number of extra bits needed to encode events from p taking a code based on q. Thus, the lower the entropy values, the more similar the two POS sequences.

A POS device designed to approach the originally observed syntax as close as possible should produce an output with a low CRE value when being compared with the original data.

Table 2. Results for baseline taggers and our tagger; Unigram tagger: assigns for each word its most probable tag and in OOV cases the overall most probable tag; Trigram: Markov tagger with linearly interpolated trigram tag history; for explanations of κ and CRE (conditional relative entropy), see text.

	accuracy	κ	CRE
Baseline Taggers:			
Unigram	89.61%	0.89	1.33
lin. interpolated Trigram	93.22%	0.93	0.61
New Tagger:			
Trigram, word repr.	96.02%	0.96	0.43

This study's tagger significantly outperforms the baseline taggers in learning the given POS patterns (two tailed McNemar test, $p = 0.001$). This improvement is also reflected in higher κ and lower CRE scores.

8 Discussion

In this paper a Markov POS tagger was introduced which benefits from automatically derived morphologic knowledge. It would be interesting to compare our approach with the ones described in the introduction, but since our training and test material is itself the output of a POS tagger, the results would

just show which tagger mimics the reference tagger best. This does not necessarily correspond to quality differences. We are aware that the same problem arises comparing our tagger with the baseline taggers above, but at least it can be noticed that our tagger learns given POS patterns better than the baseline taggers.

Furthermore, the suboptimality of training and test data may have inhibited better results. Accuracy is with high probability affected by tagging errors in the available data and is expected to increase, when cleaner data is on-hand.

References

1. Brants, T. (2000). TnT – A Statistical Part-of-Speech Tagger. In: *Proc. ANLP-2000*, 224–231, Seattle, WA.
2. Brill, E. (1995). Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. *Computational Linguistics*, 21(4):543–566.
3. Dempster, A.P., Laird, N.M., Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Statistical Society*, 39(1):1–21.
4. Jelinek, F. (1985). Markov source modeling of text generation. In Skwirzynski, J.K., editor. *The Impact of Processing Techniques on Communications*, volume E91 of *NATO ASI series*, 569–598. Dordrecht: M. Nijhoff.
5. Nascimento, M.A., da Cunha, A.C.R. (1998). An Experiment Stemming Non-Traditional Text. In *Proc. SPIRE'98*, 74–80, Santa Cruz de La Sierra, Bolivia.
6. Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In: *Proc. Conference on Empirical Methods in Natural Language Processing*, 133–142, Pennsylvania.
7. Reichel, U.D., Schiel, F. (2005). Using Morphology and Phoneme History to improve Grapheme-to-Phoneme Conversion. In *Proc. Eurospeech*, 1937–1940, Lisbon, Portugal.
8. Schiller, A., Teufel, S. (1995). *Guidelines für das Tagging deutscher Textcorpora*, <http://www.sfs.uni-tuebingen.de/Elwis/stts/stts-guide.ps.gz> (19.11.2004).
9. Schmid, H. (1995). Improvements in Part-of-Speech Tagging with an Application to German. In *EACL SIGDAT Workshop*, Dublin, Ireland. In Feldweg, Hinrichs, editors. *Lexikon und Text*, 47–50.
10. Suendermann, D., Ney, H. (2003). Synther – a New M-gram POS Tagger. In *Proc. NLP-KE*, 628–633, Bejing, China.
11. Viterbi, A.J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.
12. Voutilainen, A. (1995). A syntax-based part of speech analyser. In *Proc. of the Seventh Conference of the European Chapter of the Association for Computational Linguistics*, 157–164, Dublin, Ireland.