SYNTHER – A NEW M-GRAM POS TAGGER

David Sündermann and Hermann Ney

RWTH Aachen – University of Technology, Computer Science Department
Ahornstr. 55, 52056 Aachen, Germany
{suendermann,ney}@cs.rwth-aachen.de

ABSTRACT

In this paper, the Part-Of-Speech (POS) tagger *synther* based on m-gram statistics is described. After explaining its basic architecture, three smoothing approaches and the strategy for handling unknown words is exposed. Subsequently, *synther*'s performance is evaluated in comparison with four state-of-the-art POS taggers. All of them are trained and tested on three corpora of different languages and domains. In the course of this evaluation, *synther* resulted in the lowest error rates or at least below average error rates. Finally, it is shown that the linear interpolation smoothing strategy with coverage-dependent weights features better properties than the two other approaches.

Keywords: *synther*, (m-gram) POS tagger, linear interpolation smoothing with coverage-dependent weights, POS tagger evaluation

1. INTRODUCTION

POS taggers are used in many natural language processing tasks, e.g. in speech recognition, speech synthesis, or statistical machine translation. Their most common aim is to assign a unique POS tag to each token of the input text string.

To the best of our knowledge, statistical approaches [1], [3], [8] in most cases yield better outcomes to POS tagging than finite-state, rule-based, or memory-based approaches [2], [4]. Although the maximum entropy framework [8] seem to be the most acknowledged statistical tagging technique, it has been shown that a simple trigram approach often results in better performance [1]. As taking more context into account should improve tagging results, the usage of higher m-gram orders in conjunction with an effective smoothing method is desirable. Thus, in this paper a new approach for defining the weights of the linear interpolation smoothing technique is presented and compared with two conventional smoothing methods.

In POS tagging applications, one further viewpoint is especially considered: the handling of words which have not been seen during the training, so called out-of-vocabulary (OOV) words. In Section 4, the approach utilized within *synther* is described.

Finally, *synther*'s performance is evaluated in comparison to four state-of-the-art taggers on three corpora of different languages and domains.

2. BASIC ARCHITECTURE OF A POS TAGGER

The aim of the POS taggers (v. the schematic diagram in Figure 1) discussed in this paper is the assignment of unambiguous POS tags to the words of an input text.

Given the word (or more general token) sequence $w_1^N := w_1 \cdots w_n \cdots w_N$ on the positions $n = 1, \ldots, N$, we search for the most likely tag sequence

$$\hat{g}_1^N := \arg\max_{g_1^N} Pr(g_1^N | w_1^N).$$

Rewriting this formula by means of BAYES' law yields

$$\hat{g}_1^N = \arg\max_{g_1^N} Pr(g_1^N) \cdot Pr(w_1^N | g_1^N).$$

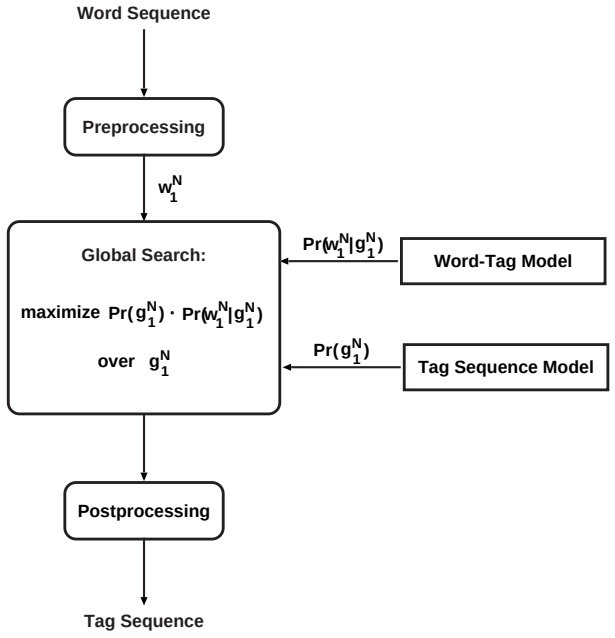

Figure 1: *Schematic Diagram of a POS Tagger*

$Pr(w_1^N|g_1^N)$ is the word-tag model and $Pr(g_1^N)$ the tag sequence model.

After factorizing the given joint probabilities into products of conditional probabilities and limiting the history which is taken into consideration, we get

$$Pr(g_1^N) = \prod_{n=1}^{N} p(g_n|g_{n-m+1}^{n-1}) \quad \text{and} \tag{1}$$

$$Pr(w_1^N|g_1^N) = \prod_{n=1}^{N} p(w_n|g_n). \tag{2}$$

3. SMOOTHING

Due to training data sparseness, the probabilities of the tag sequence model should be smoothed. Consequently, three different smoothing strategies have been tested (cf. Section 6.2). All of them use the explicit discounting approach proposed by [6] where non-zero probabilities are assigned to unseen tag sequences by discounting each frequency $N(g_{n-\mu+1}^n) > 0$ for $\mu = 2, \ldots, m$ and redistributing the probability mass. These frequencies are estimated on the basis of the tag sequence in training ${g_{tr}}_1^L$.

3.1 Absolute Discounting

In this model, the frequencies greater than zero are discounted by a constant value b_μ which is determined with the help of so-called count-counts $n_{\mu,r}$ which are the numbers of tags observed exactly r times in the training data (δ denotes the Kronecker Delta). The tag-sequence probabilities for unseen events are assigned the constant b_μ weighted by a normalization factor and the tag-sequence probability of a shortened history. Here, $N_h(g_{n-\mu+1}^{n-1})$ denotes the number of μ-grams with history $g_{n-\mu+1}^{n-1}$.

$$p(g_n|g_{n-\mu+1}^{n-1}) = \begin{cases} \frac{N(g_{n-\mu+1}^n) - b_\mu}{N(g_{n-\mu+1}^{n-1})} & \text{if } N(g_{n-\mu+1}^n) > 0 \\ b_\mu \cdot \frac{N_h(g_{n-\mu+1}^{n-1})}{N(g_{n-\mu+1}^{n-1})} \cdot \frac{p(g_n|g_{n-\mu+2}^{n-1})}{\sum_{g'} p(g'|g_{n-\mu+2}^{n-1})} & \text{otherw.} \end{cases}$$

$$b_\mu := \frac{n_{\mu,1}}{n_{\mu,1} + 2n_{\mu,2}};$$

$$n_{\mu,r} := \sum_{l=1}^{L-\mu+1} \delta(N({g_{tr}}_l^{l+\mu-1}), r) \tag{3}$$

3.2 Linear Interpolation with Weights Calculated by Means of the Leaving-One-Out Principle

This smoothing technique also uses the relative frequencies of smaller contexts down to the unigram and rates them by means of the weights λ_μ whose sum must be one. The Leaving-One-Out method discounts the non-zero frequencies of each m-gram by one, searches for the context length which results in the maximum relative frequency, and increments the according weight [1].

$$p(g_n|g_{n-m+1}^{n-1}) = \sum_{\mu=1}^{m} \lambda_\mu \frac{N(g_{n-\mu+1}^n)}{N(g_{n-\mu+1}^{n-1})}; \tag{4}$$

$$\lambda_\mu = \frac{\sum_{g_1^m \in G_\mu} N(g_1^m)}{L - m + 1} \quad \text{with}$$

$$G_\mu = \left\{ g_1^m \in \left\{ {g_{tr}}_1^m, {g_{tr}}_2^{m+1}, \ldots, {g_{tr}}_{L-m+1}^L \right\} : \right.$$

$$\left. \mu = \arg \max_{\nu=1,\ldots,m} \frac{N(g_{m-\nu+1}^m) - 1}{N(g_{m-\nu+1}^{m-1}) - 1} \right\}$$

3.3 Linear Interpolation with Weights Depending on Training Data Coverage

This technique understands that a high training data coverage for the order μ signifies that we are allowed to take more context into account and to rate that context higher. The training data coverage c_μ is the ratio between the number of different μ-grams occurred while training and that of all possible μ-grams. Hence, we have $0 \leq c_\mu \leq 1$. On the other hand, a low coverage (high sparseness) is indicator that the μ-gram weight should be curtailed.

According to the above considerations we want the interpolation weights to be positioned on a continuous function $\lambda_\mu(c_\mu)$ fulfilling the following conditions ($\hat{c}$ denotes the *optimal coverage*):

- $\lambda_\mu(0) = 0$,
- $\lambda_\mu(\hat{c}) = \max_{c_\mu} \lambda_\mu(c_\mu)$,
- $0 < \lambda_\mu(c_\mu) < \lambda_\mu(\hat{c}) \quad \text{for} \quad c_\mu \neq 0 \wedge c_\mu \neq \hat{c}$.

One simple realization of these constraints is a set of λ_μ which is computed by normalizing the values λ'_μ defined as follows. The normalization has to be executed because the sum of the interpolation weights must be unity.

$$\lambda'_\mu = \begin{cases} \dfrac{\hat{c}}{c_\mu} & \text{for} \quad c_\mu \geq \hat{c} \\ \dfrac{c_\mu}{\hat{c}} & \text{otherwise} \end{cases}$$

$\hat{c}$ should be estimated with the help of a development corpus and can be expected in the neighborhood of one percent.

4. OOV HANDLING

In Eq. (2), the word-tag probability is defined as a product of conditional probabilities $p(w|g)$ which can be derived from $p(g|w)$ by means of Bayes' law:

$$p(w|g) = \frac{p(w) \cdot p(g|w)}{p(g)} .$$

Furthermore, we understand that $p(g)$ is known and $p(w)$ constitute a factor which is equal for each possible tag sequence g_1^n and can be ignored searching for the most probable sequence. Therefore, in the following, we only discuss the estimation of the conditional probability $p(g|w)$.

In case of a word seen in training, we estimate $p(g|w)$ using relative frequencies, otherwise, we have a more detailed look at the actual word consisting of the ASCII characters $l_1, \ldots, l_I$. Especially the final characters serve as a good means to estimate word-tag probabilities of OOVs in Western European languages.

When we want to take into account relative frequencies of character strings seen in training, we have to deal with training data sparseness. Again, this leads us to the usage of smoothing strategies, cf. Section 3. *synther* uses the linear interpolation technique, v. Eq. (4), wherein the weights are defined as proposed in [9]. These considerations yield the general definition of the searched probability $p(g|w)$.

$$p(g|w) = \begin{cases} \dfrac{N(g,w)}{N(w)} & \text{for} \quad N(w) > 0 \\ \displaystyle\sum_{i=1}^{I} \lambda_i \frac{N(g, l_i, \ldots, l_I)}{N(l_i, \ldots, l_I)} & \text{otherwise} \end{cases}$$

$$\text{with} \quad \lambda_i = \frac{1}{\sigma}\left(\frac{\sigma}{1+\sigma}\right)^i$$

Here, σ is the standard deviation of the estimated tag occurrence probabilities.

5. CORPORA

In the following, the corpora used for evaluation of *synther* are compendiously presented.

Punctuation Marks (PMs) are all tokens which do not contain letters or digits. Singletons are all tokens respectively POS tags which occur only once in the training data. The m-gram perplexity is a degree of the diversity of tokens expected at each position:

$$PP_m = \left(\prod_{n=m}^{N} p(g_n | g_{n-m+1}^{n-1}) \right)^{\frac{-1}{N-m+1}} .$$

The trigram perplexity PP_3 displayed in Table 1 to 3 was computed using the linear interpolation smoothing approach explained in Section 3.3.

When we restrict the search space by exclusively taking those tags into account which have been observed in connection with the particular token, the tagging procedure can only make errors in case of either ambiguities (also OOVs) or if a token has only been seen with tags differing from that of the reference sequence. The maximum error rate ER_{max} is that error rate which we get if we always choose a wrong tag in ambiguous cases. When we randomly determine the tags according to a uniform distribution over all tags observed together with a particular word, we expect the random error rate ER_{rand}. These two error rates serve as benchmarks to assess the properties of the corpus. E.g., we note that the error rates of the POS taggers presented below are about ten percent of ER_{rand}.

5.1 Penn Treebank: Wall Street Journal Corpus

This corpus contains about one million English words of 1989 Wall Street Journal (WSJ) material with human-annotated POS tags. It was developed at the University of Pennsylvania [10], v. Table 1.

Table 1: *WSJ Corpus Statistics*

		Text	POS
Train	Sentences	43 508	
	Words + PMs	1 061 772	
	Punctuation Marks	138 279	131 075
	Vocabulary Words	46 806	45
	Vocabulary PMs	25	9
	Singletons	21 552	0
Test	Sentences	4 478	
	Words + PMs	111 220	
	OOVs	2 879 (2.6%)	0 (0%)
	Punctuation Marks	14 877 (13.4%)	14 115 (12.7%)
	PP_3	–	8.3
	ER_{max}	–	55.7%
	ER_{rand}	–	36.7%

5.2 Münster Tagging Project Corpus

This German POS tagging corpus was compiled at the University of Münster within the Münster Tagging Project (MTP). It contains articles of the newspapers *Die Zeit* and *Frankfurter Allgemeine Zeitung* [5], v. Table 2.

Table 2: *MTP Corpus Statistics*

		Text	POS
Train	Sentences	19 845	
	Words + PMs	349 699	
	Punctuation Marks	45 927	45 817
	Vocabulary Words	51 491	68
	Vocabulary PMs	27	5
	Singletons	32 678	11
Test	Sentences	2 206	
	Words + PMs	39 052	
	OOVs	3 584 (9.2%)	2 (0.0%)
	Punctuation Marks	5 125 (13.1%)	5 113 (13.1%)
	PP_3	–	7.5
	ER_{max}	–	66.7%
	ER_{rand}	–	49.8%

5.3 GENIA Corpus

The data content of the GENIA corpus is chosen from the domain of molecular biology. It is edited in American English and has been made available by the University of Tokyo [7], v. Table 3.

6. EXPERIMENTS

6.1 Evaluating *synther* in Comparison to Four Other POS Taggers

To perform an evaluation under objective conditions and to obtain comparable outcomes, *synther* has been trained and tested together with four freely available POS taggers:

- Brill's tagger based on automatically learned rules [2]
- Ratnaparkhi's maximum entropy tagger [8]
- *TnT* – a trigram tagger by Thorsten Brants [1]
- *TreeTagger* – a tagger based on decision trees provided by Helmut Schmid [11]

Table 3: *GENIA Corpus Statistics*

		Text	POS
Train	Sentences	6 191	
	Words + PMs	149 788	
	Punctuation Marks	17 181	16 900
	Vocabulary Words	11 557	56
	Vocabulary PMs	25	7
	Singletons	5 705	9
Test	Sentences	689	
	Words + PMs	16 585	
	OOVs	626 (3.8%)	0 (0%)
	Punctuation Marks	1 865 (11.2%)	1 807 (10.9%)
	PP_3	–	6.1
	ER_{max}	–	36.6%
	ER_{rand}	–	22.2%

Table 4: *POS Tagger Evaluation: Error Rates*

Corpus	Tagger	ER[%]		
		all	OOV	$\overline{\text{OOV}}$
WSJ	BRILL	3.45	17.3	3.09
	RATNAPARKHI	3.11	14.6	2.81
	synther	3.39	16.1	3.06
	TnT	3.43	14.9	3.13
	TreeTagger	3.82	30.7	3.11
MTP	BRILL	5.80	17.2	4.65
	RATNAPARKHI	5.51	12.5	4.80
	synther	5.24	13.4	4.42
	TnT	5.42	14.3	4.52
	TreeTagger	5.68	16.6	4.58
GENIA	BRILL	2.61	17.4	2.05
	RATNAPARKHI	2.03	11.5	1.66
	synther	1.94	13.2	1.50
	TnT	2.01	12.6	1.59
	TreeTagger	2.59	27.6	1.61

Table 4 shows the results of this comparison: the total error rate, that for the OOVs, and, furthermore, the outcomes exclusively for known words ($\overline{\text{OOV}}$). The latter is to separate the effect of OOV handling from that of the remaining statistics. All tests presented in this paper except for those in Section 6.2 were executed setting *synther*'s m-gram order to $m = 5$. In particular, the outcomes of Table 4 show us:

- In several cases, the m-gram statistics used by *synther* result in the lowest error rates in comparison to the other taggers tested in the course of this evaluation.

- Both BRILL's and RATNAPARKHI's POS tagger which were developed at the Department of Computer and Information Science of the University of Pennsylvania produce their best results on their in-house corpus (WSJ).

- *TnT* as well as *synther* always produce above-average outcomes. Except for the OOVs, the latter's statistics decreases the error rates by up to 6 percent relative by virtue of higher m-gram order ($m = 5$ in lieu of 3).

6.2 Comparison of Smoothing Techniques

In the introduction of this paper, we have conjectured that increasing the order of m-gram statistics should improve the tagging performance. The following test will show that this assumption is only correct if it is supported by the smoothing strategy. In Figure 2, the performance of the three smoothing approaches presented in Section 3 is displayed versus the maximum m-gram order m. These experiments are based on the WSJ corpus described in Table 1.

We note that the coverage dependent smoothing approach is the best out of these three strategies, at least for orders $m > 2$ and for the WSJ corpus. As well, this statement was confirmed on the MTP and GENIA corpus.

6.3 Influence of the Optimal Coverage Parameter $\hat{c}$ on the Smoothing Accuracy

Finally, we want to demonstrate how the accuracy of the coverage-dependent smoothing approach (cf. Section 3.3) is influenced by the optimal coverage parameter $\hat{c}$. By means of the WSJ corpus in Figure 3, we demonstrate that there is a local and also absolute minimum of the error rate curve. This minimum is located in a broad area of low gradients ($\hat{c} = 0.01 \ldots 0.1$) thus

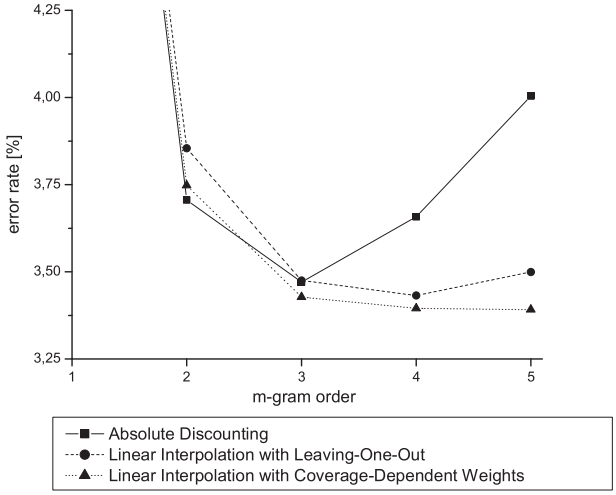

Figure 2: *Comparison of Smoothing Strategies*

determining any value within this area suffice to obtain error rates around 3.4%.

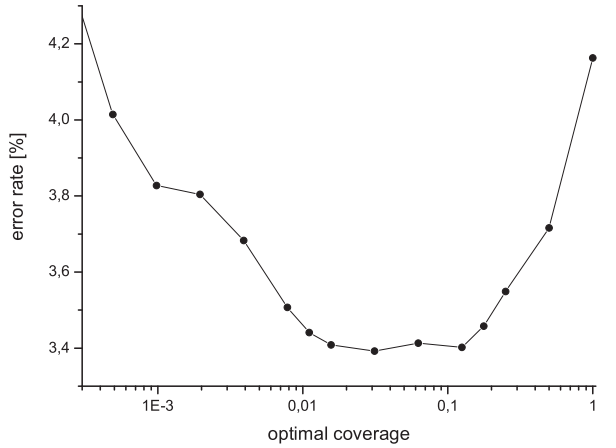

Figure 3: *Dependence of the Tagging Performance on the Optimal Coverage $\hat{c}$*

7. CONCLUSION

In this paper, we have presented the m-gram POS tagger *synther* explaining in detail its smoothing approaches and the strategy for handling unknown words. Subsequently, the new POS tagger has been evaluated on three corpora of different languages and domains and compared with four state-of-the-art taggers. We have shown that *synther* results in below-average or even the lowest error rates using a new linear interpolation smoothing technique with coverage-dependent weights.

8. REFERENCES

[1] T. Brants. 2000. TnT – A Statistical Part-of-Speech Tagger. In *Proc. of the ANLP'00*.

[2] E. Brill. 1992. A Simple Rule-Based Part of Speech Tagger. In *Proc. of the ANLP'92*.

[3] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. 1992. A Practical Part-of-Speech Tagger. In *Proc. of the ANLP'92*.

[4] W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. 1996. A Memory-Based Part-of-Speech Tagger Generator. In *Proc. of the Workshop on Very Large Corpora*.

[5] J. Kinscher and P. Steiner. 1995. Münster Tagging Projekt (MTP). *Handout for the 4th Northern German Linguistic Colloquium*.

[6] H. Ney and U. Essen. 1993. Estimating 'Small' Probabilities by Leaving-One-Out. In *Proc. of EUROSPEECH'93*.

[7] T. Ohta, Y. Tateisi, H. Mima, and J. Tsujii. 2002. GENIA Corpus: an Annotated Research Abstract Corpus in Molecular Biology Domain. In *Proc. of the HLT'02*.

[8] A. Ratnaparkhi. 1996. A Maximum Entropy Model for Part-of-Speech Tagging. In *Proc. of the EMNLP'96*.

[9] C. Samuelsson. 1996. Handling Sparse Data by Successive Abstraction. In *Proc. of the COLING'96*.

[10] B. Santorini. 1990. Part-of-Speech Tagging Guidelines for the Penn Treebank Project. *Technical Report MS-CIS-90-47, University of Pennsylvania*.

[11] H. Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proc. of the NeMLaP'94*.