New Statistical Methods for Phrase Break Prediction

Helmut Schmid and Michaela Atterer
Institute of Computational Linguistics
University of Stuttgart
Stuttgart, Germany
{schmid,atterer}@ims.uni-stuttgart.de

Abstract

The paper presents two methods for the prediction of phrase breaks. The first method uses a standard HMM part-of-speech tagger with variable context length. The second method directly encodes the distance from the last phrase break in its states. It combines the probability of a phrase break given the distance from the last phrase break with the probability of a break given the local context consisting of the surrounding words and part of speech tags. The accuracy of the new tagger is 2 percentage points higher than that of Taylor and Black (1998) on similar data.

1 Introduction

The insertion of phrase breaks is an important step on the way from raw text to synthesized speech. Phrase breaks induce prosodic structure in sentences, thus making them more naturally-sounding and intelligible. Furthermore, phrase break information is often used by other modules like accent prediction (Hirschberg, 1993; Ross and Ostendorf, 1996) or segment duration assignment (van Santen, 1994). Correct phrase break prediction is crucial for the quality of synthesized speech because a break at a wrong location can make a whole paragraph unintelligible, whereas errors in other modules (e.g. the duration module) only show local effects. Usage of break information by other modules amplifies the negative effects of break errors.

Various methods for assigning phrase breaks have been proposed, among them manually developed rules (Bachenko and Fitzpatrick, 1990), decision tree models (Wang and Hirschberg, 1992; Koehn et al., 2000), transformational rule-based learning (Fordyce and Ostendorf, 1998), memory-based learning (Marsi et al., 2003) and Hidden Markov Models (Black and Taylor, 1997; Taylor and Black, 1998). Most of the literature on prosodic phrasing agrees that, next to syntactic features, the length of the prosodic phrases plays an important role (Nespor and Vogel, 1986; Bachenko and Fitzpatrick, 1990; Ostendorf and Veilleux, 1994). Prosodic phrases tend to be balanced, such that very short and very long phrases are less likely than phrases of intermediate length. The probability of a break therefore depends to some extent on the distance from the last break. In other words, it depends on whether there was a phrase break after the preceding word or after the last but one word and so on. Modelling such sequences of hidden events is a typical task for Hidden Markov models (HMMs).

In this paper, we discuss earlier work on HMM-based phrase break prediction, describe the implementation of a phrase break tagger by means of a part-of-speech tagger, and propose a new method which directly represents the distance from the last phrase break in the state, and uses more local information (namely words in addition to POS tags).

2 Hidden Markov Models

HMM-based approaches transform phrase break prediction into a tagging task. Each word is either annotated with a break (B) or a no-break (N) tag, indicating whether a break should be placed after the word or not. This task is similar to part-of-speech tagging. Statistical POS taggers compute the most likely POS tag sequence $\hat{t}_1^n$ for a given word sequence w_1^n:

$$\begin{aligned}\hat{t}_1^n &= \arg\max_{t_1^n} p(t_1^n|w_1^n) = \arg\max_{t_1^n} \frac{p(w_1^n, t_1^n)}{p(w_1^n)} \\ &= \arg\max_{t_1^n} p(w_1^n, t_1^n)\end{aligned}$$

$p(w_1^n)$ is a constant in the maximisation (unless tokenisation is ambiguous) and therefore ignored. According to the definition of conditional probabilities, $p(w_1^n, t_1^n)$ is decomposed as follows:

$$p(w_1^n, t_1^n) = \prod_{i=1}^{n} p(t_i|w_1^{i-1}, t_1^{i-1})\; p(w_i|w_1^{i-1}, t_1^i)$$

Assuming that the next tag t_i only depends on the k preceding tags t_{i-k}^{i-1} and that the next word w_i depends only on its tag t_i, a Hidden Markov model is obtained:

$$p(w_1^n, t_1^n) = \prod_{i=1}^{n} p(t_i|t_{i-k}^{i-1})\; p(w_i|t_i) \qquad (1)$$

This is the well-known formula for HMM-based part-of-speech tagging. The best POS sequence for a given word sequence is efficiently computed by the Viterbi algorithm.

2.1 Model of Taylor and Black

Taylor and Black (1998) describe an HMM tagger which assigns phrase break tags to part-of-speech sequences. The POS tag t_i in Equation 1 is replaced by a phrase break tag and the output word w_i is replaced by the POS sequence $C_i = c_{i-M}^{i-M+L}$ (the M tags before and the $L-M$ tags after the potential phrase break, where L is the length of the POS sequence). To deal with sparse data problems, the output probabilities $p(c_{i-1}, c_i, c_{i+1}|t_i)$ are smoothed by (i) discounting frequencies with Good-Turing estimates and (ii) a back-off strategy which replaces $p(c_{i-1}, c_i, c_{i+1}|t_i)$ with $p(c_i, c_{i+1}|t_i)$ if the smoothed frequency $f_{GT}(c_i, c_{i+1}, t_i)$ is below 3.

Taylor and Black (1998) evaluated their tagger on a part of the MARSEC corpus (Knowles et al., 1996; Roach et al., 1994) which they divided into a training corpus (comprising 31,707 words and 6,346 breaks), and a test corpus (7,662 words and 1,404 breaks). They report a tagging accuracy of 91.6 %. From the accuracy and the figures for correct breaks, correct junctures and inserted junctures, it is possible to derive an f-score of 75.62 % for the prediction of phrase breaks.

From a theoretical point of view, the model of Taylor and Black (1998) is problematic because (i) the assumption that the next output symbol C_i only depends on the phrase break tag t_i, is violated due to the overlap of the POS sequence C_i with the previous sequence C_{i-1}, and (ii) the back-off smoothing strategy is incorrect because the bigram probability $p(c_i, c_{i+1}|t_i)$ is usually much higher than $p(c_{i-1}, c_i, c_{i+1}|t_i)$ and never smaller. So, replacing the probability of a POS trigram with the probability of a bigram overestimates the probability.

2.2 Using a POS tagger for phrase break prediction

The similarity of the model of Taylor and Black (1998) to POS tagging models suggests that standard POS taggers could be used for phrase break prediction. From the many POS taggers available (e.g. Brants, 2000; Ratnaparkhi, 1996), we chose the TreeTagger (Schmid, 1994) because it is based on HMMs and allows larger contexts than trigrams. The smoothing problem in Taylor and Black (1998) is solved by applying Bayes law to $p(C_i|t_i)$ and exploiting the fact that $p(C_i)$ is constant in the maximisation.

$$\begin{aligned}\hat{t}_1^n &= \arg\max_{t_1^n} \prod_{i=1}^{n} p(t_i|t_{i-k}^{i-1})\; p(C_i|t_i) \\ &= \arg\max_{t_1^n} \prod_{i=1}^{n} p(t_i|t_{i-k}^{i-1})\; \frac{p(t_i|C_i)}{p(t_i)} \qquad (2)\end{aligned}$$

$p(t_i|C_i)$ is easier to estimate than $p(C_i|t_i)$. The probabilities can be smoothed with a backoff strategy which replaces $p(t_i|c_{i-1}, c_i, c_{i+1})$ with $p(t_i|c_i, c_{i+1})$ (and potentially with $p(t_i|c_{i+1})$), if $f(c_{i-1}, c_i, c_{i+1})$ is 1 or less. Whether the tagger backs off or not depends only on the POS tags and not on the predicted phrase-break tag. A backoff factor is therefore not necessary. The backoff strategy was implemented by means of the hyphenation heuristic of the TreeTagger: The lexical probabilities of an unknown input token are replaced by the lexical probabilities of the largest suffix starting after a hyphen. If *VBD-NN* is not in the lexicon, but *NN* is, the phrase break probabilities of *NN* are used.

The number of preceding break/no-break tags on which the transition probabilities depend (i.e. the order of the HMM) is variable in this tagging approach. The input consisted of part-of-speech bigrams. Because syllables are often assumed to be a better measure of phrase length than orthographic words, we experimented with word-based and syllable-based input representations as illustrated in Figures 1 and 2.

In experiments, this tagger achieved a 2 % gain in f-score compared to Taylor and Black (1998) on similar data (see Sec. 4).

3 The New Tagger

Tagging with HMMs of high order is slow because of the large number of states and leads to data sparseness problems. The number of states and parameters decreases if the most relevant information provided by the preceding phrase break tags, namely the distance from the last

Sie	PPER-VVFIN	N
gehen	VVFIN-ADJA	N
gewagte	ADJA-NN	N
Verbindungen	NN-KON	B
und	KON-NN	N
Risiken	NN-PTKVZ	N
ein	PTKVZ-$,	B

Figure 1: Word-based input representation for the TreeTagger. (The representation is slightly modified for reasons of better illustration. The first column is not used for training and testing purposes.)

Sie	PPER-VVFIN	N
ge-	DUMMY	N
hen	VVFIN-ADJA	N
ge-	DUMMY	N
wag-	DUMMY	N
te	ADJA-NN	N
Ver-	DUMMY	N
bin-	DUMMY	N
dun-	DUMMY	N
gen	NN-KON	B
und	KON-NN	N
Ri-	DUMMY	N
si-	DUMMY	N
ken	NN-PTKVZ	N
ein	PTKVZ-$,	B

Figure 2: Syllable-based input representation for the TreeTagger. (The first column was not used for training and testing.)

phrase break, is directly encoded in the state. The distance is either measured by the number of words, or by the number of syllables. Adding more information to the local context C_i (e.g. the words around the current tagging position) could improve the accuracy.

These considerations led to the development of a new statistical phrase break tagger. Its states encode the distance from the last phrase break. They are numbered $0, 1, \ldots, D$ where D is the maximal distance considered. The "output" symbols are tuples consisting e.g. of the two preceding words and POS tags and the following word and POS tag. The new phrase break tagger computes

$$\hat{b}_1^n = \arg\max_{b_1^n} \prod_{i=1}^{n} p(b_i|d_i)p(b_i|C_i)/p(b_i) \qquad (3)$$

where $C_i = w_{i-1}^{i+1}, t_{i-1}^{i+1}$, and $d_{i+1} = |w_{i+1}|$ (length of word w_{i+1}) if $b_i = B$ (break after w_i), and $d_{i+1} = d_i + |w_{i+1}|$ if $b_i = N$ (no break after w_i). $p(B|d)$ is the probability of a phrase break d syllables (or words) after the previous phrase break, and $p(N|d) = 1 - p(B|d)$ is the probability that no phrase break occurs. The probability distribution $p(.|d)$ is the same for all distances $d \geq D$.

The Viterbi probability $\delta_{i,d}$ (i.e. the probability of the best phrase break sequence starting at the beginning of the text and ending at position i in state d) is computed as follows

$$\delta_{0,|w_1|} = 1$$

$$\delta_{0,l} = 0 \quad \text{if } l \neq |w_1|$$

$$\delta_{i,|w_{i+1}|} = \max_{d=0}^{D} \delta_{i-1,d}\, p(B|d)\, \frac{p(B|w_{i-1}^{i+1}, t_{i-1}^{i+1})}{p(B)}$$

$$\delta_{i,d+|w_{i+1}|} = \delta_{i-1,d}\, p(N|d)\, \frac{p(N|w_{i-1}^{i+1}, t_{i-1}^{i+1})}{p(N)}$$

$$\text{for } 1 \leq d < D - |w_{i+1}|$$

$$\delta_{i,D} = \max_{d=D-|w_{i+1}|}^{D} \delta_{i-1,d}\, p(N|d)\, \frac{p(N|w_{i-1}^{i+1}, t_{i-1}^{i+1})}{p(N)}$$

The approach can easily be extended for tagging with more than one type of phrase break tags.

3.1 Theoretical Background

A statistical phrase break predictor should compute the most likely phrase break tag sequence $\hat{b}_1^n$ for a given sequence of words w_1^n which is tagged with part-of-speech tags c_1^n.

$$\hat{b}_1^n = \arg\max_{b_1^n} \; p(b_1^n|w_1^n, c_1^n)$$

The probability of the phrase break tag sequence is decomposed into a product of conditional probabilities.

$$p(b_1^n|w_1^n, c_1^n) = \prod_{i=1}^{n} p(b_i|w_1^n, c_1^n, b_1^{i-1})$$

The distance d_i from the last phrase break tag is a function of b_1^{i-1} and w_1^n. Assuming that the phrase break tag b_i only depends on d_i and the local context C_i consisting of the p preceding and the f following words and part-of-speech tags, the following equation results:

$$p(b_i|w_1^n, c_1^n, b_1^{i-1}) = p(b_i|\underbrace{w_{i-p}^{i+f}, c_{i-p}^{i+f}}_{C_i}, d_i)$$

The conditional probability $p(b|C,d)$ is transformed as follows:

$$p(b|C,d) = \frac{p(b,C,d)}{p(C,d)} = p(b|d)\, p(C|b,d) \frac{p(d)}{p(C,d)}$$

$$= p(b|d)p(C|b)\frac{p(d)}{p(C,d)}\frac{p(C,d|b)}{p(C|b)p(d|b)}$$

$$= p(b|d)\frac{p(b|C)}{p(b)}\frac{p(C)p(d)}{p(C,d)}\frac{p(C,d|b)}{p(C|b)p(d|b)}$$

The expression $F = \frac{p(C)\,p(d)}{p(C,d)}\frac{p(C,d|b)}{p(C|b)\,p(d|b)}$ is 1 if the statistical dependence between C and d is identical to the statistical dependence between C and d given b. Computation of this factor on real data indeed often returned values close to 1, but values as high as 3 or as low as 0.2 occurred, as well. In order to simplify the model and to avoid sparse data problems caused by the huge number of these factors, we neglect them. This step is also motivated by the similarity of the resulting formula to the POS tagging formula (Eq. 2). The best sequence of phrase break tags is obtained by the maximisation in Equation 3, which is just the computation performed by the proposed tagger.

3.2 Syllable Counts

Measuring the phrase length by the number of syllables requires a counting method for syllables. We approximate the number of syllables with the number of vowels and diphtongs if a word contains vowels, and with the number of characters otherwise. This simple heuristic produced results virtually as good as those obtained with a sophisticated lexicon-based method.

3.3 Parameter Estimation

The transition probabilities $p(b|d)$ are directly estimated with relative frequency estimates for distances up to D, where $D + 1$ is the first value for which the relative frequency estimate is undefined (i.e. the smallest unobserved phrase length). Smoothing turned out to be unnecessary for these parameters.

The "lexical" probabilities $p(b|c_1^n)$ are smoothed by adding the weighted backoff probabilities $\beta\ p(b|c_2^n)$ to the frequency counts $f(b, c_1^n)$ (see Eq. 4). The backoff probabilities have been smoothed in the same way. The elements of the local context c_1^n are ordered according to increasing relevance (cf. Table 2). The optimal order was initially guessed and later confirmed in experiments.

$$\hat{f}(b, c_i^n) = f(b, c_i^n) + \beta\ p(b|c_{i+1}^n) \qquad (4)$$

$$p(b|c_i^n) = \frac{\hat{f}(b, c_i^n)}{\sum_{b'} \hat{f}(b', c_i^n)} \qquad (5)$$

3.4 Log-Linear Weights

The phrase break tagging formula of Equation 3 assigns equal weight to the contextual probabilities $p(b|d)$ and the local probabilities $p(b|C)$ although the local probabilities seem more important. In order to apply a weighting to the different factors of our model, we turned it into a log-linear model by means of the equation:

$$\prod_{i=1}^{n} p(b_i|d_i)\frac{p(b_i|C_i)}{p(b_i)} = e^{\sum_{i=1}^{n} \alpha_i}$$

where

$$\alpha_i = log\, p(b_i|d_i) + log\, p(b_i|C_i) - log\, p(b_i)$$

Multiplying the logarithms of the probabilities with λ weights and renormalising the scores by a constant Z, a log-linear model is obtained.

$$p(t_1^n|o_1^n) = \frac{1}{Z}\, e^{\sum_{i=1}^{n} \alpha_i}$$

where

$$\alpha_i = \lambda_1 log\, p(b_i|d_i) + \lambda_2 log\, p(b_i|C_i) - \lambda_3 log\, p(b_i)$$

4 Evaluation

The phrase break predictors were evaluated on the English MARSEC corpus with 52,000 words and two German corpora, a 7,400-word Radio News corpus (RNC) and a 90,000 word newspaper corpus (NPC). The MARSEC is recordings from the BBC transcribed by two experts on the basis of an auditory analysis (Knowles et al., 1996). The RNC is recorded radio news from 1995 which was manually annotated with prosodic labels (cf. Mayer, 1995; Rapp, 1998). The NPC is a subcorpus of the Negra treebank (Skut et al., 1998), which was manually annotated with phrase breaks according to the method described in (Hirschberg and Prieto, 1996). Average phrase lengths were 5.02 (MARSEC), 4.93 (NPC) and 4.53 (RNC) words. The standard deviation was 2.60 (MARSEC), 2.54 (NPC) and 2.12 (RNC). We calculated recall (percentage of breaks in the corpus which were predicted), precision (percentage of correctly predicted breaks) and the f-score as $2 * precision * recall/(precision + recall)$.

Figures 3, 4 and 5 show the results obtained by the TreeTagger as described in Section 2.2 using POS-bigrams. We investigated context lengths (i.e. Break/Non-break n-grams) of up to 24 syllables and up to 12 words respectively.

The pruning parameter of the TreeTagger was optimized on the test data for each context length. In the plots, the labelling of the x-axis refers to the syllable-based model. The values for the word-based model are plotted in the ratio 1:2. (One word is represented as 2 syllables in the plot. This approximates the actual word-to-syllable ratio in the corpora.) Overall, the syllable-based input representation tends to be better than the word-based input representation.

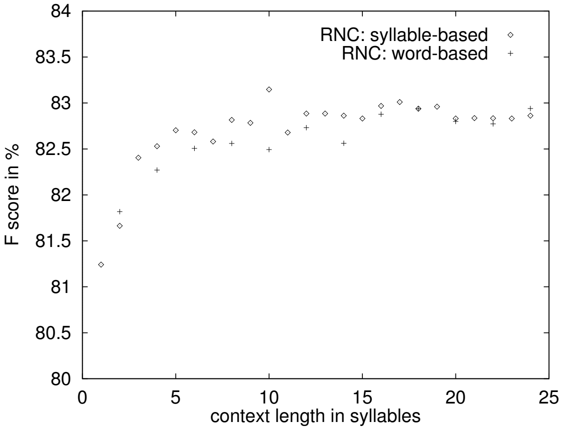

Figure 3: F-scores for TreeTagger on the radio news corpus (RNC).

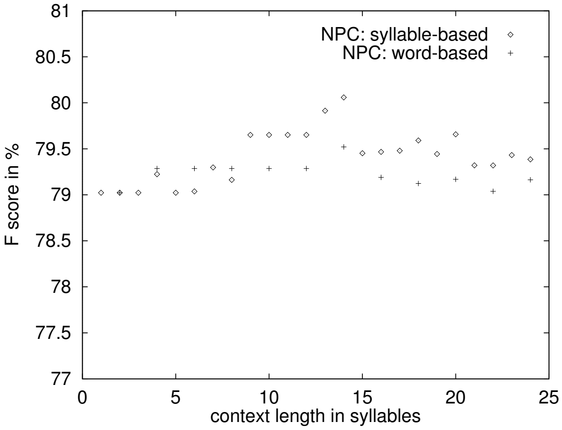

Figure 4: F-scores for TreeTagger on the newspaper corpus (NPC).

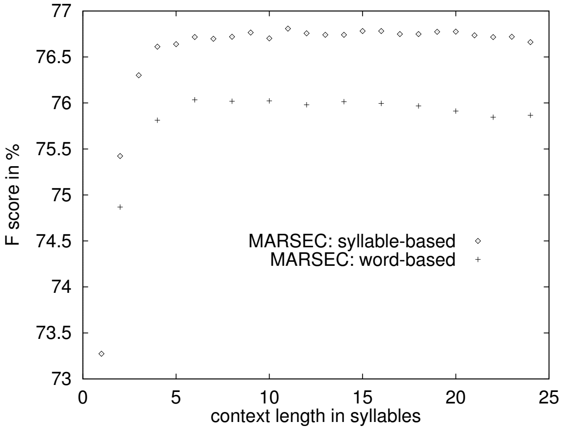

Figure 5: F-scores for TreeTagger on the MARSEC corpus.

PB-Tagger	precision	recall	f-score
NPC	80.05	80.41	80.23
MARSEC	76.35	79.30	77.80
RNC	84.41	86.10	85.25
TreeTagger	precision	recall	f-score
NPC	89.02	72.74	80.06
MARSEC	75.32	78.35	76.81
RNC	82.18	84.14	83.15
Baseline	precision	recall	f-score
NPC	81.85	61.58	70.28
MARSEC	86.86	51.79	64.89
RNC	95.36	49.95	65.56
Upper limit	precision	recall	f-score
RNC	84.96	85.74	85.35

Table 1: Evaluation results

Table 1 compares the TreeTagger results with those obtained with the new statistical phrase break tagger. The evaluation is based on ten-fold cross validation. The TreeTagger results were obtained with optimal parameter settings, whereas the smoothing parameter of the new tagger was optimised with nine-fold cross-validation on the training data inside the ten-fold cross validation loop. Nevertheless, the new tagger is better on all corpora.

Table 1 also shows baseline results and an upper limit for comparison. The baseline was obtained by placing phrase breaks at punctuation positions[1]. In order to determine an upper limit, we measured how well multiple pronunciations of the same text agreed in the placement of phrase breaks. The scores were computed in the same way as with automatically tagged data. Repeated pronunciations were only available for part of the RNC data and there is some uncertainty in the upper limit scores due to the small size of the data (2807 tokens overall).

The MARSEC results were obtained with a version of the corpus that was automatically tagged with POS tags using the TreeTagger[2].

[1] We considered periods, question and exclamation marks, commas, colons, semicolons, parentheses and quotation marks as punctuation.

[2] The TreeTagger achieves 96.5 % tagging accuracy on

Other results reported later were obtained with the original POS tags of the MARSEC corpus. A comparison with the results of (Taylor and Black, 1998) (f-score 75.62 %) is difficult because it is not clear which part of the MARSEC corpus they used and how it was divided into test and training data.

Figure 6 shows how the f-score depends on the smoothing parameter β. For $\beta > 4$, the variance of the f-score was small for the MARSEC and the NPC corpus, whereas the smaller RNC corpus shows a higher variance.

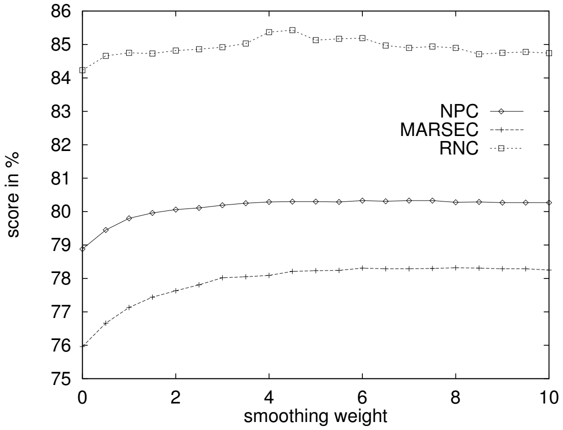

Figure 6: Variation of smoothing weight β

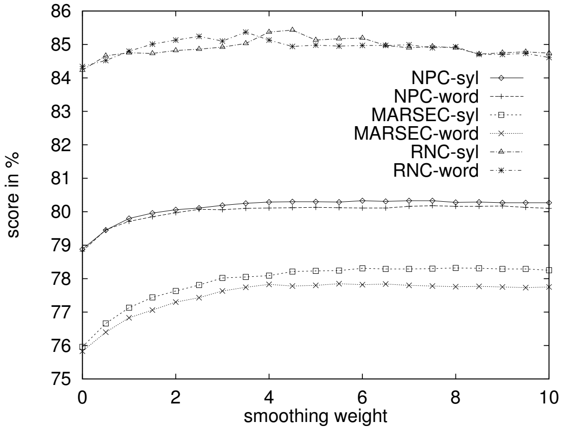

Figure 7: Word-based vs. syllable-based distance

Figure 7 compares word-based and syllable-based distance measures, showing a small, but quite consistent advantage for the syllable-based measure on the MARSEC and the NPC corpus and mixed results on the RNC corpus.

The results of an experiment with log-linear weights (see Sec. 3.4) are summarised in Figure 8. We achieved a small f-score gain (0.2 % for MARSEC and 0.1 % for NPC) with a λ_2 value of 1.3 (with λ_1 and λ_3 set to 1). In all other experiments, the λ values were 1.

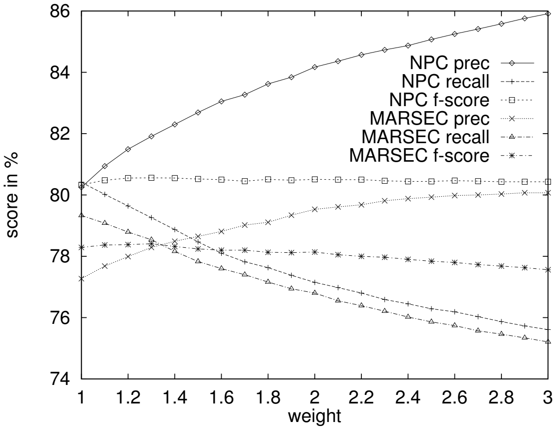

Figure 8: Variation of log-linear weight λ_2

Finally, we investigated on the NPC corpus how much additional local context contributed to the results. Table 2 shows precision, recall and f-score values for different contexts. Adding the following word w_{+1} to the POS trigram improved the f-score by 0.6 %. So, adding words indeed helped the tagger.

context	Prec.	Recall	F-Score
$w_{-1}\,w\,t_{-1}\,w_{+1}\,t\,t_{+1}$	80.18	80.49	80.33
$w\,t_{-1}\,w_{+1}\,t\,t_{+1}$	80.16	80.48	80.32
$t_{-1}\,w_{+1}\,t\,t_{+1}$	80.15	80.43	80.29
$w_{+1}\,t\,t_{+1}$	79.69	80.25	79.97
$t\,t_{+1}$	79.31	79.00	79.15
t_{+1}	79.06	69.39	73.91
$t_{-1}\,t\,t_{+1}$	79.34	80.01	79.67

Table 2: variation of local context

The TreeTagger processed about 20000 tokens per second on a Sun Blade 1000 with 750 Mhz CPU. The new tagger was implemented in Perl and processed about 1000 tokens per second. We would expect a C implementation to have similar speed as the TreeTagger because they are both based on HMMs of similar complexity.

5 Summary

We improved the HMM-based phrase break tagging method of Taylor and Black (1998) by using a better smoothing technique, larger N-grams and syllable-based input representations.

Furthermore, we presented a new statistical method for phrase-break prediction which directly encodes the distance from the last phrase break in its state and combines two types of

the Penn treebank with ten-fold cross validation.

conditional probabilities, namely (i) the probability of the next phrase break tag given the distance from the preceding phrase break and (ii) the probability of the next break tag given the surrounding words and part-of-speech tags. The accuracy on the MARSEC corpus measured by the f-score is more than 2 percentage points higher than that obtained by Taylor and Black (1998) on the same corpus using an unknown splitting into training and test data. With a German corpus, we were able to show that the tagging accuracy comes close to the upper limit defined by the agreement between different pronunciations of the same text.

References

Bachenko, J. and Fitzpatrick, E. (1990). A computational grammar of discourse-neutral prosodic phrasing in english. *Computational Linguistics*, 16(3):155–170.

Black, A. W. and Taylor, P. (1997). Assigning phrase breaks from part-of-speech sequences. In *Eurospeech*, pages 995–998.

Brants, T. (2000). Tnt - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000*, Seattle, WA.

Fordyce, C. and Ostendorf, M. (1998). Prosody prediction for speech synthesis using transformational rule-based learning. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*.

Hirschberg, J. (1993). Pitch accent in context: Predicting intonational prominence from text. *Artificial Intelligence*, 63.

Hirschberg, J. and Prieto, P. (1996). Training intonational phrasing automatically for english and spanish text-to-speech. *Speech Communication*, 18:281–290.

Knowles, G., Williams, B., and Taylor, L. (1996). *A Corpus of Formal British English Speech: The Lancaster/IBM Spoken English Corpus*. Longman, London.

Koehn, P., Abney, S., Hirschberg, J., and Collins, M. (2000). Improving intonational phrasing with syntactic information. In *ICASSP 2000*.

Marsi, E., Reynaert, M., van den Bosch, A., Daelemans, W., and Hoste, V. (2003). Learning to predict pitch accents and prosodic boundaries in Dutch. In *Proceedings of the 41th Annual Meeting of the ACL*, Sapporo, Japan.

Mayer, J. (1995). Transcription of german intonation – the stuttgart system. Technical report, Institute for Computational Linguistics, University of Stuttgart.

Nespor, M. and Vogel, I. (1986). *Prosodic Phonology*. Foris publications.

Ostendorf, M. and Veilleux, N. (1994). A hierarchical stochastic model for automatic predictions of prosodic boundary location. *Computational Linguistics*, 20(1).

Rapp, S. (1998). *Automatische Erstellung von Korpora für die Prosodieforschung*. PhD thesis, Institute for Computational Linguistics, University of Stuttgart, Stuttgart, Germany.

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*. University of Pennsylvania.

Roach, P., Knowles, G., Varadi, T., and Arnfield, S. (1994). MARSEC: a machine-readable spoken english corpus. *Journal of the International Phonetic Association*, 24(24):47–53.

Ross, K. and Ostendorf, M. (1996). Prediction of abstract prosodic labels for speech synthesis. *Computer Speech and Language*.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Skut, W., Brants, T., Krenn, B., and Uszkoreit, H. (1998). A linguistically interpreted corpus of german newspaper text. In *Proceedings of the 10th European Summer School in Logic, Language and Information (ESSLLI'98), Workshop on Recent Advances in Corpus Annotation*.

Taylor, P. and Black, A. W. (1998). Assigning phrase breaks from part-of-speech sequences. *Computer, Speech and Language*, 12(2):99–117.

van Santen, J. P. (1994). Assignment of segmental duration in text-to-speech synthesis. *Computer Speech and Language*.

Wang and Hirschberg (1992). Automatic classification of intonational phrase boundaries. *Computer Speech and Language*, 6:175–196.