# Predicting phrase breaks with memory-based learning

*Bertjan Busser, Walter Daelemans, Antal van den Bosch*

ILK / Computational Linguistics
Tilburg University, The Netherlands
{g.j.busser,antalb}@kub.nl, daelem@uia.ua.ac.be

Proceedings 4th ISCA Tutorial and Research Workshop on Speech Synthesis, 2001

## Abstract

We investigate whether Memory-Based Learning (MBL) can be used to predict Phrase Breaks (PBs) in speech production reliably. The MBL approach is compared to the HMM approach described in [Taylor and Black, 1998] using the same corpus and information sources. We show that a simple memory-based learning algorithm that uses only minimal context and information outperforms the HMM approach, in terms of precision and recall. An exhaustive search of variants of algorithms, metrics and information sources does not bring any significant further improvement.

## 1. Introduction

In this study, we investigate whether Memory-Based Learning (MBL) can be used to reliably predict Phrase Breaks (PBs) in speech production. PBs are perceived as pauses in speech by the listener, and can be interpreted as marking boundaries between prosodic constituents.

Memory-Based Learning (MBL) is a classification-based, supervised learning approach. In this framework, a problem is treated as a classification task: given a set of feature values describing the context in which a PB appears and any other relevant information as input, a *classifier* selects the appropriate output class from a finite number of a priori given classes – here, marking the absence or presence of a break in a specific position within a string of words.

This paper compares the MBL approach to the HMM approach described in [Taylor and Black, 1998] using the same corpus and information sources. We extend the accuracy results reported in [Taylor and Black, 1998] with the more informative measures of precision and recall of boundaries, and show that a simple memory-based learning algorithm using only minimal context and information outperforms the HMM approach. An exhaustive search of variants of algorithms, metrics and information sources does not bring any significant further improvement.

The paper is structured as follows: Section 2 gives a brief overview of memory-based learning, and Section 3 describes how phrase prediction is defined as a learnable classification task. In Section 4 the results of the experiments are given, and in Section 5 the results of the experiments, as well as the comparison with [Taylor and Black, 1998] are discussed.

## 2. Memory-Based Learning

MBL keeps all training data in memory and only abstracts at classification time by extrapolating a class from the most similar item(s) in memory. In earlier work [Daelemans et al., 1999] we have shown that for typical natural language processing tasks this learning approach is at an advantage, because it remembers exceptional, low-frequency cases which are nevertheless useful to extrapolate from. In contrast, "eager" machine learning methods (such as decision tree learners and rule inducers) forget this useful information, because of their pruning and frequency-based abstraction methods. Moreover, the automatic feature weighting in the similarity metric of a memory-based learner makes the approach well-suited for domains with large numbers of features from homogeneous or heterogeneous sources, as it embodies a smoothing-by-similarity method when data is sparse [Zavrel and Daelemans, 1997]. For our experiments we have used TiMBL[1], an MBL software package developed in our group [Daelemans et al., 2000]. In addition, we have obtained especially good results in the past with MBL applied to the tasks of word-level phonemisation (grapheme-to-phoneme conversion) and stress assignment for different languages [Daelemans and Van den Bosch, 1996, Van den Bosch, 1997, Busser et al., 1999]. In sum, MBL is a natural candidate for use in predicting phonological properties at sentence level.

The TiMBL software emulates the following variants of MBL:

IB1: The distance between a test item and each memory item is defined as the number of features for which they have a different value (overlap metric) [Aha et al., 1991].

IB1-IG: In most cases, not all features are equally relevant for solving the task; this variant uses information gain (an information-theoretic notion measuring the reduction of uncertainty about the class to be predicted when knowing the value of a feature) to weight the cost of a feature value mismatch during comparison [Daelemans and Van den Bosch, 1992].

IB1-MVDM: For typical symbolic (nominal) features, values are not ordered. In the previous variants, mismatches between values are all interpreted as equally important, regardless of how similar (in terms of classification behaviour) the values are. We adopted the *modified value difference metric* [Cost and Salzberg, 1993] to assign a different distance between each pair of values of the same feature.

MVDM-IG: MVDM with IG weighting.

IGTREE: In this variant, an oblivious decision tree is created with features as tests, and ordered according to information gain of features, as a heuristic approximation of the computationally more expensive pure MBL variants [Daelemans et al., 1997].

## 3. Task Description

In this Section, we describe the task and the information sources available for the data used, and show how the task was formu-

[1] TiMBL is available from: http://ilk.kub.nl/.

| Tag | function |
|---|---|
| cc | conjunction |
| dt | determiner |
| ex | exsitential "there" |
| in | Preposition |
| md | modal |
| of | of |
| pdt | predeterminer |
| pos | possessive |
| rp | particle |
| to | to |
| wdt | wh-determiner |
| wp | wh-pronoun |
| wrb | wh-adverb |

Table 1: List of the closed class POS tags that together make up the F (function word) tag in the CFP tag set.

lated as a classification task, making it learnable by MBL.

The experiments in this study are performed on the 'MAchine Readable Spoken English Corpus' (MARSEC), semi-automatically annotated for PBs by Taylor and Black (see [Taylor and Black, 1998]). The corpus consists of 40 stories, 39369 words, and 7780 PBs. Taylor and Black designated 30 stories to be the 'train set', and 10 stories as 'test set'.

### 3.1. Information sources

In the MARSEC corpus two types of information are provided: the words of the text themselves, and their Part-Of-Speech (POS) tags, as well as the location of PBs. The POS tags were added by [Taylor and Black, 1998], using a HMM tagger trained on the Wall Street Journal [Marcus et al., 1993]. From these two sources we produce two additional types of information. First, the **CFP-value**, which distinguishes between Content words (C), Function words (F), and Punctuation (P) based on a list of POS tags. Table 1 lists the closed-class POS tags that make up the "F" class.

Second, we introduce the **expanded tag**, which contains the word itself if it is a function word, and its POS tag otherwise. Both types of features change the level of granularity of pure POS and word features; the CFP representation brings the granularity of POS tagging down to three symbols, while the second type of features removes all open-class word values and leaves a fixed number of POS tags and closed-class words as values. In sum, we provide the MBL learner with maximally four sources of information, or features, for each word: the word itself, its POS tag, its CFP-value, and an expanded tag.

As MBL (like other machine learning systems) needs fixed-width feature vectors as input, we apply a windowing approach to extract such vectors from each sentence in the corpus. For each position between two words in each sentence, we create a feature vector with the information sources available for a fixed number of words to the left and to the right of that position. The class corresponding to each feature vector is either 1 when there is a boundary at that position, or 0 otherwise.

For example, the following sentence (taken from the first sentence of section a01 of the MARSEC corpus), using a context size of 2 words to the left and 2 words to the right (referred to in the remainder of this paper as a 2-2 window width), would result in feature vectors such as shown in Table 2. The bars in the example indicate phrase breaks.

| left_2 | left_1 | right_1 | right_2 | correct answer |
|---|---|---|---|---|
| EMPTY | More | news | about | 0 |
| More | news | about | the | 0 |
| news | about | the | Reverend | 0 |
| about | the | Reverend | Sun | 0 |
| the | Reverend | Sun | Myung | 0 |
| Reverend | Sun | Myung | Moon | 0 |
| Sun | Myung | Moon | COMMA | 0 |
| Myung | Moon | COMMA | founder | 0 |
| Moon | COMMA | founder | of | 1 |
| COMMA | founder | of | the | 0 |

Table 2: Example feature vectors with a 2-2 width (two words to the left and to the right of a position in the sentence).

| window type | left_2 | left_1 | right_1 | right_2 | correct answer |
|---|---|---|---|---|---|
| words | the | Reverend | Sun | Myung | 0 |
| tags | dt | nnp | nnp | nnp | 0 |
| cfp | f | c | c | c | 0 |
| expanded tags | the | nnp | nnp | nnp | 0 |

Table 3: Examples of feature vectors with different types of information.

> More news about the Reverend Sun Myung Moon , | founder of the Unification church , | who 's currently in jail | for tax evasion : || he was awarded an honorary degree last week | by the Roman Catholic University .

Instead of, or in addition to, the word in each context position, we can provide the information sources described earlier (CFP feature, POS tag, expanded tag) to test which type or which combination of information works best. See Table 3 for examples of windows containing different features. It is possible to mix information from different sources in one instance as long as the feature vectors are fixed-length with the same feature always referring to the same type of information. For a 2-2 width incorporating all available information, this would result in a feature vector with 16 features (4 information sources for each of the four context positions).

## 4. Experiments

First, we convert the results reported in [Taylor and Black, 1998] to the more standard evaluative metrics of precision, recall, and F-score. In a second set of experiments, we try to optimize the MBL metrics, algorithm selection, context width and information sources, using cross-validation on the training set.

### 4.1. Converting Taylor and Black's results to precision and recall

In this experiment, we compare the simplest memory-based learning to the results of [Taylor and Black, 1998] using the same training and test data. Although they used Hidden Markov Modelling (HMM), their goal was the same: to predict PBs. Taylor and Black report on several variants (see Table 4). They distinguish experiments on the basis of type of POS model and two values of $n$ in the n-grams used in the Phrase Break model. The POS model is defined by whether the method for prediction is deterministic (non-HMM) or probabilistic (HMM), and tagset used; in this case either punctuation, non-punctuation (P) or punctuation, function word, content word (our CFP).

The first row in Table 4, Det P, uses a simple deterministic algorithm that places a break after punctuation, determined by the underlying POS HMM trained on the Wall Street Journal tagged corpus [Marcus et al., 1993]. The second row, Det PCF, is also deterministic, and places a break either after punctuation, or after a content word that is followed by a function word. The POS model here categorizes input as punctuation, content word, or function word. The third row, Prob P-1, is the first that uses an HMM phrase break model based on 1-grams representing punctuation or non-punctuation. The fourth row, Prob P-6, uses a sequence of 6 punctuation/non-punctuation markers as a Phrase Break model. Here it becomes possible for the system to allow context to determine the placement of breaks. The fifth and sixth rows are analoguous to the third and fourth rows, but use the reduced tagset of {punctuation, content word, function word} (CFP) rather than the dichotomous {punctuation, non-punctuation}.

With respect to the measures Taylor and Black use, they report that they prefer the 'Breaks Correct' measure over the 'Junctures Correct' measure as it takes into account that breaks are a minority class. However, there are other measures that fullfill the same goal but are used much more frequent. We feel that using precision and recall, and their harmonic mean $F_\beta$ [Rijsbergen, 1979], is preferable with respect to comparability and clarity.

The most straightforward and informative way to evaluate the behavior of a system on this task is in terms of recall (how many PBs occurring in the test data are correctly predicted) and precision (how many of the predicted PBs are correct according to the test data). The so-called $F_\beta$ score [Rijsbergen, 1979], with $\beta = 1$, is an harmonic mean of both precision and recall, and is a good measure of overall quality.

However, to evaluate the performance of their method, Taylor and Black use the percentage of breaks correct, percentage of junctures correct, and percentage of juncture insertions. They are defined as follows (from [Taylor and Black, 1998], p. 6):

$$breaks\ correct = b.c. = \frac{B - D - S}{B}\ (*100) \quad (1)$$

$$junctures\ correct\ = j.c. = \frac{N - D - I - S}{N}\ (*100) \quad (2)$$

$$juncture\ insertions = j.i. = \frac{I}{N}\ (*100) \quad (3)$$

where N is the total number of word boundaries or junctures (6772, [Taylor and Black, 1998], p. 4), B is the total number of breaks predicted, D is the number of deletions, or PBs that should have been predicted but were not, I is the number of insertions, i.e. where no PB should be predicted but was anyway, and S is the number of substitutions, which is only relevant when predicting break levels, i.e. when a distinction is made between a minor and a major break (or a short and a long pause, respectively). Rewriting these formulas towards precision and recall consists of the following steps:

$b = number\ of\ correct\ breaks = 1404$
([Taylor and Black, 1998], $p.4$)

$B = b - D + I$

$I = N * \frac{j.i.}{100\%}\ \ (from Eqn.3)$

$D = N * (1 - \frac{j.c.}{100\%} - \frac{j.i.}{100\%})\ \ (from Eqn.2)$

$precision = \frac{B-I}{B}$

$recall = \frac{b-D}{b} = \frac{B-I}{b}\ \ (via Eqn.5)$

$F_\beta = \frac{(\beta^2+1)*precision*recall}{(\beta^2*precision)+recall}$
([Rijsbergen, 1979])

The results of this conversion are listed in Table 4. The first three columns are copied from ([Black, 1998], p. 7). The last six are the conversion results added by us. Note that there is an error in the results reported in [Taylor and Black, 1998] on the "Det PCF" model (second row); their numbers result in a converted recall above 100%.

### 4.2. Cross-validation memory-based learning

The goal of this experiment is to create an unbiased, full experimental matrix covering all variations of information sources available to the learner, as well as all algorithmic parameter settings. Available information sources are word, POS tag, CFP value, and expanded tag (word for function words, POS tag otherwise). This information can be made available for zero or more (two in this experiment) positions to the left or right of a focus position for which a class (boundary or not) has to be predicted. MBL algorithm parameters investigated here are gain ratio weighting or no weighting, the value of $k$ (number of nearest neighbors used to extrapolate from), and simple overlap or MVDM. See Section 2. We constructed an experimental matrix, varying over all these factors. In each cell of the matrix, we cross-validated on the training set using leave-one-out (each training item in turn is used as a validation item with all other training items as training set).

Table 5 lists the scores on the original test set for the ten best information source and parameter setting combinations in this experimental matrix obtained by cross-validation on the training set. In other words, the ranking is based on the $F_\beta$ score on leave-one-out experiments on the training material. A higher $F_\beta$ on training material appears to correspond with a higher $F_\beta$ on test material. We see that limited information about limited context, namely information about the POS tag one word to the left and to the right, scores highest. One of the two best-scoring variations does not even use feature weighting; it assigns equal weight to both left and right POS tags.

When experimenting further with higher values of $k$ than one for the top performing settings, no significant further improvement was noticed. Surprisingly, although the different information sources have different a priori relevance (as measured e.g. by their mutual information with the class to be predicted, see Appendix A), using this more fine-grained information does not help in solving the task any better. Moreover, parameter and feature selection optimization using standard techniques like forward and backward search did not improve upon these results

| Exp | b.c.(%) | j.c(%) | j.i(%) | I | D | B | Prec(%) | Rec(%) | $F_\beta$(%) |
|---|---|---|---|---|---|---|---|---|---|
| Det P | 54.3 | 90.8 | 0.9 | 65.1 | 642.8 | 826.3 | 92.1 | 54.1 | 68.2 |
| Det PCF | 84.4 | 71.3 | 31.7 | 2431.2 | -231.4 | 4066.5 | 40.2 | 116.5 | 59.8 |
| Prob P-1 | 55.0 | 91.1 | 0.8 | 61.3 | 620.6 | 844.7 | 92.7 | 55.8 | 69.7 |
| Prob P-6 | 58.6 | 88.0 | 5.4 | 413.0 | 505.7 | 1311.3 | 68.5 | 64.0 | 66.2 |
| Prob PCF-1 | 54.9 | 91.1 | 0.8 | 61.3 | 620.6 | 844.7 | 92.7 | 55.8 | 69.7 |
| Prob PCF-6 | 68.3 | 89.4 | 5.9 | 448.2 | 364.7 | 1487.5 | 69.9 | 74.0 | 71.9 |

Table 4: Results reported by Taylor and Black, and conversions to intermediate values, precision, recall, and $F_\beta$ (see text for explanation).

| Weights | Metric | Left context | Right context | Info | $F_\beta$ | precision | recall | accuracy |
|---|---|---|---|---|---|---|---|---|
| none | overlap | 1 | 1 | Tag | 74.4 | 76.1 | 72.8 | 90.0 |
| GR | overlap | 1 | 1 | Tag | 74.4 | 76.1 | 72.8 | 90.0 |
| GR | mvdm | 1 | 1 | Tag | 74.3 | 76.0 | 72.8 | 89.9 |
| none | mvdm | 1 | 1 | Tag | 74.3 | 76.0 | 72.8 | 89.9 |
| GR | overlap | 2 | 1 | Tag | 74.1 | 77.8 | 70.7 | 90.1 |
| none | overlap | 2 | 1 | Tag | 73.7 | 78.3 | 69.6 | 90.0 |
| GR | overlap | 2 | 2 | Tag | 73.5 | 77.6 | 69.8 | 89.9 |
| none | mvdm | 2 | 1 | Tag | 73.4 | 77.1 | 70.1 | 89.9 |
| GR | mvdm | 2 | 1 | Tag | 73.4 | 77.1 | 70.0 | 89.8 |
| GR | overlap | 1 | 2 | Tag | 72.8 | 76.0 | 69.9 | 89.6 |

Table 5: Top 10 scores in the experimental matrix.

## 5. Conclusions

The best $F_\beta$-value Taylor and Black achieve is listed in the bottom row of Table 4: 71.9%. In this study, all variants listed in top10s are better; the *lowest* $F_\beta$-value listed is 72.8% (Table 5, bottom row).

From these results we can conclude that although the HMM approach yields better accuracy, the MBL approach seems to be better at predicting PBs, which is what we were interested in. Also, these results are in a sense discouraging, as we have not been able to show that more complex information sources (making possible more fine-grained representations of the linguistic contexts fed to the PB-predictor) improve the accuracy achieved with very limited context and very general information. This seems to us to suggest that not a lot is actually learned by the HMM and MBL machine learning algorithms. We can show that learning techniques like HMM and MBL can improve upon the most naive statistical baselines: for example, never guessing a break yields an accuracy of 81.7% (all 1405 breaks are missed, which is an accuracy error of 18.3%), but naturally this score is associated with a recall of 0% and a non-existant precision and $F_\beta$. The HMM and MBL approaches are in a sense themselves only baselines. It seems to be the case that with the information sources used in the Taylor and Black study and in this study, we lack essential information for learning the PB prediction task properly. Whether more information will emerge simply from having more data (i.e., allowing word features to become better predictors), or whether intrinsically different types of features are necessary, remains a topic for further research.

## 6. References

[Aha et al., 1991] Aha, D. W., Kibler, D., and Albert, M. (1991). Instance-based learning algorithms. *Machine Learning*, 6:37–66.

[Busser et al., 1999] Busser, G., Daelemans, W., and Van den Bosch, A. (1999). Machine learning of word pronunciation: the case against abstraction. In *Proceedings of the Sixth European Conference on Speech Communication and Technology, Eurospeech99, Budapest, Hungary*, pages 2123–2126.

[Cost and Salzberg, 1993] Cost, S. and Salzberg, S. (1993). A weighted nearest neighbour algorithm for learning with symbolic features. *Machine Learning*, 10:57–78.

[Daelemans and Van den Bosch, 1992] Daelemans, W. and Van den Bosch, A. (1992). Generalisation performance of backpropagation learning on a syllabification task. In Drossaers, M. F. J. and Nijholt, A., editors, *Proc. of TWLT3: Connectionism and Natural Language Processing*, pages 27–37, Enschede. Twente University.

[Daelemans and Van den Bosch, 1996] Daelemans, W. and Van den Bosch, A. (1996). Language-independent data-oriented grapheme-to-phoneme conversion. In Van Santen, J. P. H., Sproat, R. W., Olive, J. P., and Hirschberg, J., editors, *Progress in Speech Processing*, pages 77–89. Springer-Verlag, Berlin.

[Daelemans et al., 1997] Daelemans, W., Van den Bosch, A., and Weijters, A. (1997). IGTree: using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11:407–423.

[Daelemans et al., 1999] Daelemans, W., Van den Bosch, A., and Zavrel, J. (1999). Forgetting exceptions is harmful in language learning. *Machine Learning, Special issue on Natural Language Learning*, 34:11–41.

[Daelemans et al., 2000] Daelemans, W., Zavrel, J., Van der Sloot, K., and Van den Bosch, A. (2000). TiMBL: Tilburg Memory Based Learner, version 3.0, reference manual. Technical Report ILK-0001, ILK, Tilburg University.

[Marcus et al., 1993] Marcus, M., Santorini, B., and Marcinkiewicz, M. (1993). Building a large annotated corpus of english: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

[Rijsbergen, 1979] Rijsbergen, C. v. (1979). *Information retrieval*. Buttensworth, London.

[Taylor and Black, 1998] Taylor, P. and Black, A. (1998). Assigning phrase breaks from part of speech sequences. *Computer Speech and Language*.

[Van den Bosch, 1997] Van den Bosch, A. (1997). *Learning to pronounce written words: A study in inductive language learning*. PhD thesis, Universiteit Maastricht.

[Zavrel and Daelemans, 1997] Zavrel, J. and Daelemans, W. (1997). Memory-based learning: Using similarity for smoothing. In *Proc. of 35th annual meeting of the ACL*, Madrid.

# A. Appendix A

Feature names are systematic, and consist of 3 symbols. The first is a letter (either L or R), and indicates wether this feature occurs to the left or to the right of the word boundary being analyzed. The second is a digit and tells hom far to the right or left the feature is from the current word boundary. The last symbol is a letter (W,T,C, or X) which indicates what type of information is contained in the feature: word, tag, CFP, or expanded tag.

| Feature number | Feature name | Example value | GR |
|---|---|---|---|
| 1 | L5W | Reverend | 0.015121 |
| 2 | L5T | NNP | 0.000745 |
| 3 | L5C | C | 0.001349 |
| 4 | L5X | | 0.00137 |
| 5 | L4W | Sun | 0.01559 |
| 6 | L4T | NNP | 0.00154 |
| 7 | L4C | C | 0.00230 |
| 8 | L4X | | 0.00236 |
| 9 | L3W | Myung | 0.01663 |
| 10 | L3T | NNP | 0.00356 |
| 11 | L3C | C | 0.00425 |
| 12 | L3X | NNP | 0.00430 |
| 13 | L2W | Moon | 0.02304 |
| 14 | L2T | NNP | 0.01022 |
| 15 | L2C | C | 0.01463 |
| 16 | L2X | NNP | 0.00970 |
| 17 | L1W | , | 0.04580 |
| 18 | L1T | PUNC | 0.06401 |
| 19 | L1C | P | 0.17408 |
| 20 | L1X | PUNC | 0.05541 |
| 21 | R1W | founder | 0.03658 |
| 22 | R1T | NN | 0.05108 |
| 23 | R1C | C | 0.08997 |
| 24 | R1X | NN | 0.04650 |
| 25 | R2W | of | 0.02128 |
| 26 | R2T | OF | 0.01371 |
| 27 | R2C | F | 0.01824 |
| 28 | R2X | of | 0.01302 |
| 29 | R3W | the | 0.01768 |
| 30 | R3T | DT | 0.00403 |
| 31 | R3C | F | 0.00721 |
| 32 | R3X | the | 0.00419 |
| 33 | R4W | Unification | 0.01578 |
| 34 | R4T | NNP | 0.00145 |
| 35 | R4C | C | 0.00190 |
| 36 | R4X | NNP | 0.00203 |

Table 6: Feature weights for all four types of features (word, tag, CFP, extended), with example values taken from the first sentence of the corpus.