Enhanced word decomposition by calibrating the decision threshold of probabilistic models and using a model ensemble

Sebastian Spiegler
Intelligent Systems Laboratory,
University of Bristol, U.K.
spiegler@cs.bris.ac.uk

Peter A. Flach
Intelligent Systems Laboratory,
University of Bristol, U.K.
peter.flach@bristol.ac.uk

Abstract

This paper demonstrates that the use of ensemble methods and carefully calibrating the decision threshold can significantly improve the performance of machine learning methods for morphological word decomposition. We employ two algorithms which come from a family of generative probabilistic models. The models consider segment boundaries as hidden variables and include probabilities for letter transitions within segments. The advantage of this model family is that it can learn from small datasets and easily generalises to larger datasets. The first algorithm PROMODES, which participated in the Morpho Challenge 2009 (an international competition for unsupervised morphological analysis) employs a lower order model whereas the second algorithm PROMODES-H is a novel development of the first using a higher order model. We present the mathematical description for both algorithms, conduct experiments on the morphologically rich language Zulu and compare characteristics of both algorithms based on the experimental results.

1 Introduction

Words are often considered as the smallest unit of a language when examining the grammatical structure or the meaning of sentences, referred to as syntax and semantics, however, words themselves possess an internal structure denominated by the term *word morphology*. It is worthwhile studying this internal structure since a language description using its morphological formation is more compact and complete than listing all possible words. This study is called *morphological analysis*. According to Goldsmith (2009) four tasks are assigned to morphological analysis: *word decomposition* into morphemes, building *morpheme dictionaries*, defining *morphosyntactical rules* which state how morphemes can be combined to valid words and defining *morphophonological rules* that specify phonological changes morphemes undergo when they are combined to words. Results of morphological analysis are applied in speech synthesis (Sproat, 1996) and recognition (Hirsimaki et al., 2006), machine translation (Amtrup, 2003) and information retrieval (Kettunen, 2009).

1.1 Background

In the past years, there has been a lot of interest and activity in the development of algorithms for morphological analysis. All these approaches have in common that they build a *morphological model* which is then applied to analyse words. Models are constructed using *rule-based methods* (Mooney and Califf, 1996; Muggleton and Bain, 1999), *connectionist methods* (Rumelhart and McClelland, 1986; Gasser, 1994) or *statistical* or *probabilistic* methods (Harris, 1955; Hafer and Weiss, 1974). Another way of classifying approaches is based on the *learning aspect* during the construction of the morphological model. If the data for training the model has the same structure as the desired output of the morphological analysis, in other words, if a morphological model is learnt from labelled data, the algorithm is classified under *supervised learning*. An example for a supervised algorithm is given by Oflazer et al. (2001). If the input data has no information towards the desired output of the analysis, the algorithm uses *unsupervised learning*. Unsupervised algorithms for morphological analysis are Linguistica (Goldsmith, 2001), Morfessor (Creutz, 2006) and Paramor (Monson, 2008). *Minimally or semi-supervised algorithms* are provided with partial information during the learning process. This

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 375–383,
Uppsala, Sweden, 11-16 July 2010.

has been done, for instance, by Shalonova et al. (2009) who provided stems in addition to a word list in order to find multiple pre- and suffixes. A comparison of different levels of supervision for morphology learning on Zulu has been carried out by Spiegler et al. (2008).

Our two algorithms, PROMODES and PROMODES-H, perform *word decomposition* and are based on *probabilistic methods* by incorporating a probabilistic generative model.[1] Their parameters can be estimated from either labelled data, using maximum likelihood estimates, or from unlabelled data by expectation maximization[2] which makes them either *supervised* or *unsupervised* algorithms.

The purpose of this paper is an analysis of the underlying probabilistic models and the types of errors committed by each one. Furthermore, it is investigated how the decision threshold can be calibrated and a model ensemble is tested.

The remainder is structured as follows. In Section 2 we introduce the probabilistic generative process and show in Sections 2.1 and 2.2 how we incorporate this process in PROMODES and PROMODES-H. We start our experiments with examining the learning behaviour of the algorithms in 3.1. Subsequently, we perform a position-wise comparison of predictions in 3.2, show how we find a better decision threshold for placing morpheme boundaries in 3.3 and combine both algorithms using a model ensemble to leverage individual strengths in 3.4. In 3.5 we examine how the single algorithms contribute to the result of the ensemble. In Section 4 we will compare our approaches to related work and in Section 5 we will draw our conclusions.

2 Probabilistic generative model

Intuitively, we could say that our models describe the process of word generation from the left to the right by alternately using two dice, the first for deciding whether to place a morpheme boundary in the current word position and the second to get a corresponding letter transition. We are trying to reverse this process in order to find the underlying sequence of tosses which determine the morpheme boundaries. We are applying the notion of a *probabilistic generative process* consisting of words as observed variables X and their hidden segmentation as latent variables Y. If a generative model is fully parameterised it can be reversed to find the underlying word decomposition by forming the conditional probability distribution $Pr(Y|X)$.

Let us first define the model-independent components. A given *word* $w_j \in W$ with $1 \leq j \leq |W|$ consists of n letters and has $m = n-1$ positions for inserting boundaries. A word's segmentation is depicted as a *boundary vector* $b_j = (b_{j1}, \ldots, b_{jm})$ consisting of *boundary values* $b_{ji} \in \{0,1\}$ with $1 \leq i \leq m$ which disclose whether or not a boundary is placed in position i. A *letter* $l_{j,i\text{-}1}$ precedes the position i in w_j and a letter l_{ji} follows it. Both letters $l_{j,i\text{-}1}$ and l_{ji} are part of an alphabet. Furthermore, we introduce a *letter transition* t_{ji} which goes from $l_{j,i\text{-}1}$ to l_{ji}.

2.1 PROMODES

PROMODES is based on a zero-order model for boundaries b_{ji} and on a first-order model for letter transitions t_{ji}. It describes a word's segmentation by its morpheme boundaries and resulting letter transitions within morphemes. A boundary vector b_j is found by evaluating each position i with

$$\begin{aligned} &\arg\max_{b_{ji}} Pr(b_{ji}|t_{ji}) = \\ &\arg\max_{b_{ji}} Pr(b_{ji})Pr(t_{ji}|b_{ji}) \quad . \end{aligned} \tag{1}$$

The first component of the equation above is the *probability distribution over non-/boundaries* $Pr(b_{ji})$. We assume that a boundary in i is inserted independently from other boundaries (zero-order) and the graphemic representation of the word, however, is conditioned on the length of the word m_j which means that the probability distribution is in fact $Pr(b_{ji}|m_j)$. We guarantee $\sum_{r=0}^{1} Pr(b_{ji}{=}r|m_j) = 1$. To simplify the notation in later explanations, we will refer to $Pr(b_{ji}|m_j)$ as $Pr(b_{ji})$.

The second component is the *letter transition probability distribution* $Pr(t_{ji}|b_{ji})$. We suppose a first-order Markov chain consisting of transitions t_{ji} from letter $l_{j,i\text{-}1} \in A_{\mathscr{B}}$ to letter $l_{ji} \in A$ where A is a regular letter alphabet and $A_{\mathscr{B}}{=}A \cup \{\mathscr{B}\}$ includes $\mathscr{B}$ as an abstract morpheme start symbol which can occur in $l_{j,i\text{-}1}$. For instance, the suffix '*s*' of the verb form *gets*, marking *3rd person singular*, would be modelled as $\mathscr{B} \rightarrow s$ whereas a morpheme internal transition could be $g \rightarrow e$. We

[1] PROMODES stands for PRObabilistic MOdel for different DEgrees of Supervision. The H of PROMODES-H refers to Higher order.

[2] In (Spiegler et al., 2009; Spiegler et al., 2010a) we have presented an unsupervised version of PROMODES.

guarantee $\sum_{l_{ji} \in A} Pr(t_{ji}|b_{ji})=1$ with t_{ji} being a transition from a certain $l_{j,i-1} \in A_{\mathscr{B}}$ to l_{ji}. The advantage of the model is that instead of evaluating an exponential number of possible segmentations (2^m), the best segmentation $b^*_j=(b^*_{j1},\ldots,b^*_{jm})$ is found with $2m$ position-wise evaluations using

$$b^*_{ji} = \arg\max_{b_{ji}} Pr(b_{ji}|t_{ji}) \quad (2)$$

$$= \begin{cases} 1, \text{ if } Pr(b_{ji}=1)Pr(t_{ji}|b_{ji}=1) \\ \quad > Pr(b_{ji}=0)Pr(t_{ji}|b_{ji}=0) \\ 0, \text{ otherwise } . \end{cases}$$

The simplifying assumptions made, however, reduce the expressive power of the model by not allowing any dependencies on preceding boundaries or letters. This can lead to over-segmentation and therefore influences the performance of PROMODES. For this reason, we have extended the model which led to PROMODES-H, a higher-order probabilistic model.

2.2 PROMODES-H

In contrast to the original PROMODES model, we also consider the boundary value $b_{j,i\text{-}1}$ and modify our transition assumptions for PROMODES-H in such a way that the new algorithm applies a first-order boundary model and a second-order transition model. A transition t_{ji} is now defined as a transition from an abstract symbol in $l_{j,i\text{-}1} \in \{\mathscr{N},\mathscr{B}\}$ to a letter in $l_{ji} \in A$. The abstract symbol is $\mathscr{N}$ or $\mathscr{B}$ depending on whether b_{ji} is 0 or 1. This holds equivalently for letter transitions $t_{j,i\text{-}1}$. The suffix of our previous example *gets* would be modelled $\mathscr{N} \rightarrow t \rightarrow \mathscr{B} \rightarrow s$.

Our boundary vector b_j is then constructed from

$$\arg\max_{b_{ji}} Pr(b_{ji}|t_{ji},t_{j,i\text{-}1},b_{j,i\text{-}1}) = \quad (3)$$
$$\arg\max_{b_{ji}} Pr(b_{ji}|b_{j,i\text{-}1})Pr(t_{ji}|b_{ji},t_{j,i\text{-}1},b_{j,i\text{-}1}) \ .$$

The first component, the *probability distribution over non-/boundaries* $Pr(b_{ji}|b_{j,i\text{-}1})$, satisfies $\sum_{r=0}^{1} Pr(b_{ji}=r|b_{j,i\text{-}1})=1$ with $b_{j,i\text{-}1}, b_{ji} \in \{0,1\}$. As for PROMODES, $Pr(b_{ji}|b_{j,i\text{-}1})$ is shorthand for $Pr(b_{ji}|b_{j,i\text{-}1},m_j)$. The second component, the *letter transition probability distribution* $Pr(t_{ji}|b_{ji},b_{j,i\text{-}1})$, fulfils $\sum_{l_{ji} \in A} Pr(t_{ji}|b_{ji},t_{j,i\text{-}1},b_{j,i\text{-}1})=1$ with t_{ji} being a transition from a certain $l_{j,i-1} \in A_{\mathscr{B}}$ to l_{ji}. Once again, we find the word's best segmentation b^*_j in $2m$ evaluations with

$$b^*_{ji} = \arg\max_{b_{ji}} Pr(b_{ji}|t_{ji},t_{j,i\text{-}1},b_{j,i\text{-}1}) = \quad (4)$$

$$\begin{cases} 1, \text{ if } Pr(b_{ji}=1|b_{j,i\text{-}1})Pr(t_{ji}|b_{ji}=1,t_{j,i\text{-}1},b_{j,i\text{-}1}) \\ \quad > Pr(b_{ji}=0|b_{j,i\text{-}1})Pr(t_{ji}|b_{ji}=0,t_{j,i\text{-}1},b_{j,i\text{-}1}) \\ 0, \text{ otherwise } . \end{cases}$$

We will show in the experimental results that increasing the memory of the algorithm by looking at $b_{j,i-1}$ leads to a better performance.

3 Experiments and Results

In the Morpho Challenge 2009, PROMODES achieved competitive results on Finnish, Turkish, English and German – and scored highest on non-vowelized and vowelized Arabic compared to 9 other algorithms (Kurimo et al., 2009). For the experiments described below, we chose the South African language *Zulu* since our research work mainly aims at creating morphological resources for under-resourced indigenous languages. Zulu is an *agglutinative language* with a complex morphology where multiple prefixes and suffixes contribute to a word's meaning. Nevertheless, it seems that segment boundaries are more likely in certain word positions. The PROMODES family harnesses this characteristic in combination with describing morphemes by letter transitions. From the *Ukwabelana corpus* (Spiegler et al., 2010b) we sampled 2500 Zulu words with a single segmentation each.

3.1 Learning with increasing experience

In our first experiment we applied 10-fold cross-validation on datasets ranging from 500 to 2500 words with the goal of measuring how the learning improves with increasing experience in terms of training set size. We want to remind the reader that our two algorithms are aimed at small datasets.

We randomly split each dataset into 10 subsets where each subset was a test set and the corresponding 9 remaining sets were merged to a training set. We kept the labels of the training set to determine model parameters through maximum likelihood estimates and applied each model to the test set from which we had removed the answer keys. We compared results on the test set against the ground truth by counting true positive (TP), false positive (FP), true negative (TN) and

false negative (FN) *morpheme boundary* predictions. Counts were summarised using precision[3], recall[4] and f-measure[5], as shown in Table 1.

Data	Precision	Recall	F-measure
500	0.7127±0.0418	0.3500±0.0272	0.4687±0.0284
1000	0.7435±0.0556	0.3350±0.0197	0.4614±0.0250
1500	0.7460±0.0529	0.3160±0.0150	0.4435±0.0206
2000	0.7504±0.0235	0.3068±0.0141	0.4354±0.0168
2500	0.7557±0.0356	0.3045±0.0138	0.4337±0.0163

(a) PROMODES

Data	Precision	Recall	F-measure
500	0.6983±0.0511	0.4938±0.0404	0.5776±0.0395
1000	0.6865±0.0298	0.5177±0.0177	0.5901±0.0205
1500	0.6952±0.0308	0.5376±0.0197	0.6058±0.0173
2000	0.7008±0.0140	0.5316±0.0146	0.6044±0.0110
2500	0.6941±0.0184	0.5396±0.0218	0.6068±0.0151

(b) PROMODES-H

Table 1: 10-fold cross-validation on Zulu.

For PROMODES we can see in Table 1a that the precision increases slightly from 0.7127 to 0.7557 whereas the recall decreases from 0.3500 to 0.3045 going from dataset size 500 to 2500. This suggests that to some extent fewer morpheme boundaries are discovered but the ones which are found are more likely to be correct. We believe that this effect is caused by the limited memory of the model which uses order zero for the occurrence of a boundary and order one for letter transitions. It seems that the model gets quickly saturated in terms of incorporating new information and therefore precision and recall do not drastically change for increasing dataset sizes. In Table 1b we show results for PROMODES-H. Across the datasets precision stays comparatively constant around a mean of 0.6949 whereas the recall increases from 0.4938 to 0.5396. Compared to PROMODES we observe an increase in recall between 0.1438 and 0.2351 at a cost of a decrease in precision between 0.0144 and 0.0616.

Since both algorithms show different behaviour with increasing experience and PROMODES-H yields a higher f-measure across all datasets, we will investigate in the next experiments how these differences manifest themselves at the boundary level.

[3] $precision = \frac{TP}{TP+FP}$.
[4] $recall = \frac{TP}{TP+FN}$.
[5] $f\text{-}measure = \frac{2 \cdot precision \cdot recall}{precision+recall}$.

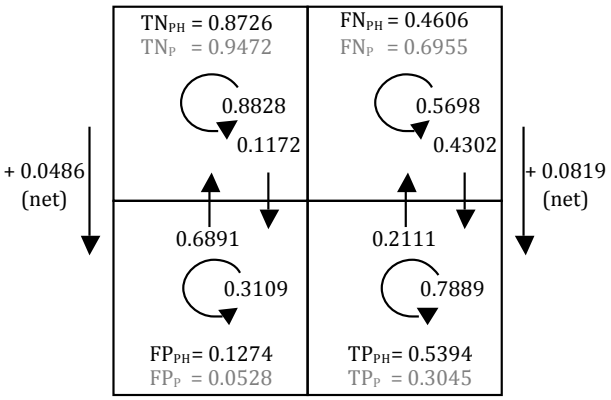


Figure 1: Contingency table for PROMODES [grey with subscript *P*] and PROMODES-H [black with subscript *PH*] results including gross and net changes of PROMODES-H.

3.2 Position-wise comparison of algorithmic predictions

In the second experiment, we investigated which aspects of PROMODES-H in comparison to PROMODES led to the above described differences in performance. For this reason we broke down the summary measures of precision and recall into their original components: true/false positive (TP/FP) and negative (TN/FN) counts presented in the 2×2 contingency table of Figure 1. For general evidence, we averaged across all experiments using relative frequencies. Note that the relative frequencies of positives (TP + FN) and negatives (TN + FP) each sum to one.

The goal was to find out how predictions in each word position changed when applying PROMODES-H instead of PROMODES. This would show where the algorithms agree and where they disagree. PROMODES classifies non-boundaries in 0.9472 of the times correctly as TN and in 0.0528 of the times falsely as boundaries (FP). The algorithm correctly labels 0.3045 of the positions as boundaries (TP) and 0.6955 falsely as non-boundaries (FN). We can see that PROMODES follows a rather conservative approach.

When applying PROMODES-H, the majority of the FP's are turned into non-boundaries, however, a slightly higher number of previously correctly labelled non-boundaries are turned into false boundaries. The net change is a 0.0486 increase in FP's which is the reason for the decrease in precision. On the other side, more false non-

boundaries (FN) are turned into boundaries than in the opposite direction with a net increase of 0.0819 of correct boundaries which led to the increased recall. Since the deduction of precision is less than the increase of recall, a better over-all performance of PROMODES-H is achieved.

In summary, PROMODES predicts more accurately non-boundaries whereas PROMODES-H is better at finding morpheme boundaries. So far we have based our decision for placing a boundary in a certain word position on Equation 2 and 4 assuming that $P(b_{ji}{=}1|\ldots) > P(b_{ji}{=}0|\ldots)$[6] gives the best result. However, if the underlying distribution for boundaries given the evidence is skewed, it might be possible to improve results by introducing a certain decision threshold for inserting morpheme boundaries. We will put this idea to the test in the following section.

3.3 Calibration of the decision threshold

For the third experiment we slightly changed our experimental setup. Instead of dividing datasets during 10-fold cross-validation into training and test subsets with the ratio of 9:1 we randomly split the data into training, validation and test sets with the ratio of 8:1:1. We then run our experiments and measured contingency table counts.

Rather than placing a boundary if $P(b_{ji}{=}1|\ldots) > P(b_{ji}{=}0|\ldots)$ which corresponds to $P(b_{ji}{=}1|\ldots) > 0.50$ we introduced a decision threshold $P(b_{ji}{=}1|\ldots) > h$ with $0 \leq h \leq 1$. This is based on the assumption that the underlying distribution $P(b_{ji}|\ldots)$ might be skewed and an optimal decision can be achieved at a different threshold. The optimal threshold was sought on the validation set and evaluated on the test set. An overview over the validation and test results is given in Table 2. We want to point out that the threshold which yields the best f-measure result on the validation set returns almost the same result on the separate test set for both algorithms which suggests the existence of a general optimal threshold.

Since this experiment provided us with a set of data points where the recall varied monotonically with the threshold and the precision changed accordingly, we reverted to *precision-recall curves* (PR curves) from machine learning. Following Davis and Goadrich (2006) the algorithmic performance can be analysed more informatively using these kinds of curves. The PR curve is plotted with recall on the x-axis and precision on the y-axis for increasing thresholds h. The PR curves for PROMODES and PROMODES-H are shown in Figure 2 on the validation set from which we learnt our optimal thresholds h^*. Points were connected for readability only – points on the PR curve cannot be interpolated linearly.

In addition to the PR curves, we plotted isometrics for corresponding f-measure values which are defined as $precision = \frac{f\text{-}measure \cdot recall}{2recall - f\text{-}measure}$ and are hyperboles. For increasing f-measure values the isometrics are moving further to the top-right corner of the plot. For a threshold of $h = 0.50$ (marked by '$\diamond$') PROMODES-H has a better performance than PROMODES. Nevertheless, across the entire PR curve none of the algorithms dominates. One curve would *dominate* another if all data points of the dominated curve were beneath or equal to the dominating one. PROMODES has its optimal threshold at $h^* = 0.36$ and PROMODES-H at $h^* = 0.37$ where PROMODES has a slightly higher f-measure than PROMODES-H. The points of optimal f-measure performance are marked with '$\triangle$' on the PR curve.

	Prec.	**Recall**	**F-meas.**
PROMODES validation (h=0.50)	0.7522	0.3087	0.4378
PROMODES test (h=0.50)	0.7540	0.3084	0.4378
PROMODES validation (h^*=0.36)	0.5857	0.7824	0.6699
PROMODES test (h^*=0.36)	0.5869	0.7803	0.6699
PROMODES-H validation (h=0.50)	0.6983	0.5333	0.6047
PROMODES-H test (h=0.50)	0.6960	0.5319	0.6030
PROMODES-H validation (h^*=0.37)	0.5848	0.7491	0.6568
PROMODES-H test (h^*=0.37)	0.5857	0.7491	0.6574

Table 2: PROMODES and PROMODES-H on validation and test set.

Summarizing, we have shown that both algorithms commit different errors at the word position level whereas PROMODES is better in predicting non-boundaries and PROMODES-H gives better results for morpheme boundaries at the default threshold of $h = 0.50$. In this section, we demonstrated that across different decision thresholds h for $P(b_{ji}{=}1|\ldots) > h$ none of algorithms dominates the other one, and at the optimal threshold PROMODES achieves a slightly higher performance than PROMODES-H. The question which arises is whether we can combine PROMODES and PROMODES-H in an *ensemble* that leverages individual strengths of both.

[6] Based on Equation 2 and 4 we use the notation $P(b_{ji}|\ldots)$ if we do not want to specify the algorithm.

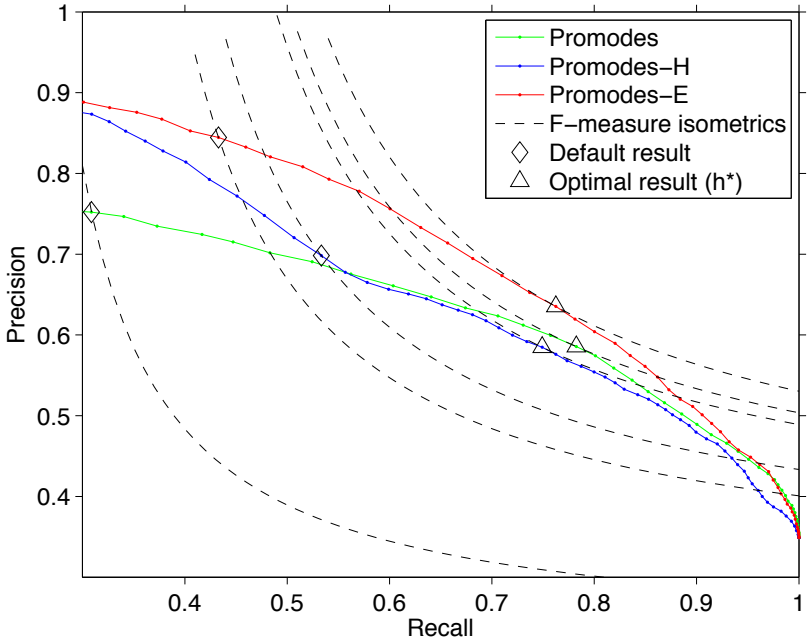


Figure 2: Precision-recall curves for algorithms on validation set.

3.4 A model ensemble to leverage individual strengths

A *model ensemble* is a set of individually trained classifiers whose predictions are combined when classifying new instances (Opitz and Maclin, 1999). The idea is that by combining PROMODES and PROMODES-H, we would be able to avoid certain errors each model commits by consulting the other model as well. We introduce PROMODES-E as the *ensemble* of PROMODES and PROMODES-H. PROMODES-E accesses the individual probabilities $Pr(b_{ji}{=}1|\dots)$ and simply averages them:

$$\frac{Pr(b_{ji}{=}1|t_{ji}) + Pr(b_{ji}{=}1|t_{ji}, b_{j,i\text{-}1}, t_{j,i\text{-}1})}{2} > h \quad .$$

As before, we used the default threshold $h = 0.50$ and found the calibrated threshold $h^* = 0.38$, marked with '◇' and '△' in Figure 2 and shown in Table 3. The calibrated threshold improves the f-measure over both PROMODES and PROMODES-H.

	Prec.	Recall	F-meas.
PROMODES-E validation (h=0.50)	0.8445	0.4328	0.5723
PROMODES-E test (h=0.50)	0.8438	0.4352	0.5742
PROMODES-E validation (h^*=0.38)	0.6354	0.7625	0.6931
PROMODES-E test (h^*=0.38)	0.6350	0.7620	0.6927

Table 3: PROMODES-E on validation and test set.

The optimal solution applying $h^* = 0.38$ is more balanced between precision and recall and boosted the original result by 0.1185 on the test set. Compared to its components PROMODES and PROMODES-H the f-measure increased by 0.0228 and 0.0353 on the test set.

In short, we have shown that by combining PROMODES and PROMODES-H and finding the optimal threshold, the ensemble PROMODES-E gives better results than the individual models themselves and therefore manages to leverage the individual strengths of both to a certain extend. However, can we pinpoint the exact contribution of each individual algorithm to the improved result? We try to find an answer to this question in the analysis of the subsequent section.

3.5 Analysis of calibrated algorithms and their model ensemble

For the entire dataset of 2500 words, we have examined boundary predictions dependent on the relative word position. In Figure 3 and 4 we have plotted the absolute counts of correct boundaries (TP) and non-boundaries (TN) which PROMODES predicted but not PROMODES-H, and vice versa, as continuous lines. We furthermore provided the number of individual predictions which were ultimately adopted by PROMODES-E in the ensemble as dashed lines.

In Figure 3a we can see for the default threshold that PROMODES performs better in predicting non-boundaries in the *middle* and the *end* of the word in comparison to PROMODES-H. Figure 3b

shows the statistics for correctly predicted boundaries. Here, PROMODES-H outperforms PROMODES in predicting correct boundaries across the *entire* word length. After the calibration, shown in Figure 4a, PROMODES-H improves the correct prediction of non-boundaries at the *beginning* of the word whereas PROMODES performs better at the *end*. For the boundary prediction in Figure 4b the signal disappears after calibration.

Concluding, it appears that our test language Zulu has certain features which are modelled best with either a lower or higher-order model. Therefore, the ensemble leveraged strengths of both algorithms which led to a better overall performance with a calibrated threshold.

4 Related work

We have presented two probabilistic generative models for word decomposition, PROMODES and PROMODES-H. Another generative model for morphological analysis has been described by Snover and Brent (2001) and Snover et al. (2002), however, they were interested in finding *paradigms* as sets of mutual exclusive operations on a word form whereas we are describing a generative process using morpheme boundaries and resulting letter transitions.

Moreover, our probabilistic models seem to resemble *Hidden Markov Models* (HMMs) by having certain states and transitions. The main difference is that we have dependencies between states as well as between emissions whereas in HMMs emissions only depend on the underlying state.

Combining different morphological analysers has been performed, for example, by Atwell and Roberts (2006) and Spiegler et al. (2009). Their approaches, though, used *majority vote* to decide whether a morpheme boundary is inserted in a certain word position or not. The algorithms themselves were treated as *black-boxes*.

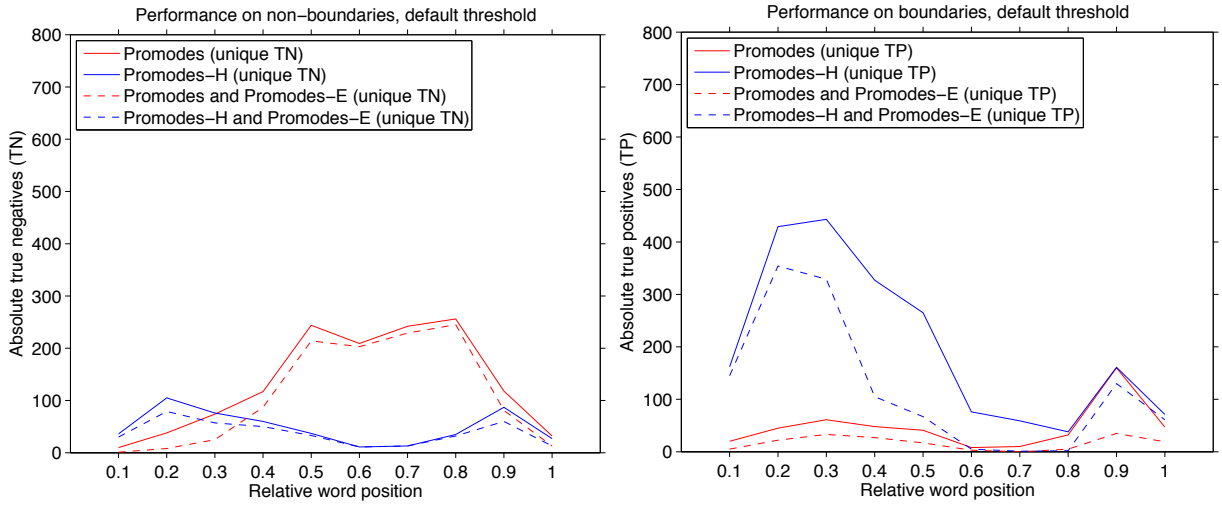
Monson et al. (2009) described an indirect approach to probabilistically combine ParaMor (Monson, 2008) and Morfessor (Creutz, 2006). They used a natural language tagger which was trained on the output of ParaMor and Morfessor. The goal was to mimic each algorithm since ParaMor is rule-based and there is no access to Morfessor's internally used probabilities. The tagger would then return a probability for starting a new morpheme in a certain position based on the original algorithm. These probabilities in combination with a threshold, learnt on a different dataset, were used to merge word analyses. In contrast, our *ensemble* algorithm PROMODES-E directly accesses the probabilistic framework of each algorithm and combines them based on an optimal threshold learnt on a validation set.

5 Conclusions

We have presented a method to learn a *calibrated decision threshold* from a validation set and demonstrated that *ensemble methods* in connection with calibrated decision thresholds can give better results than the individual models themselves. We introduced two algorithms for word decomposition which are based on generative probabilistic models. The models consider segment boundaries as hidden variables and include probabilities for letter transitions within segments. PROMODES contains a lower order model whereas PROMODES-H is a novel development of PROMODES with a higher order model. For both algorithms, we defined the mathematical model and performed experiments on language data of the morphologically complex language *Zulu*. We compared the performance on increasing training set sizes and analysed for each word position whether their boundary prediction agreed or disagreed. We found out that PROMODES was better in predicting non-boundaries and PROMODES-H gave better results for morpheme boundaries at a default decision threshold. At an optimal decision threshold, however, both yielded a similar f-measure result. We then performed a further analysis based on relative word positions and found out that the calibrated PROMODES-H predicted non-boundaries better for initial word positions whereas the calibrated PROMODES for mid- and final word positions. For boundaries, the calibrated algorithms had a similar behaviour. Subsequently, we showed that a *model ensemble* of both algorithms in conjunction with finding an optimal threshold exceeded the performance of the single algorithms at their individually optimal threshold.

Acknowledgements

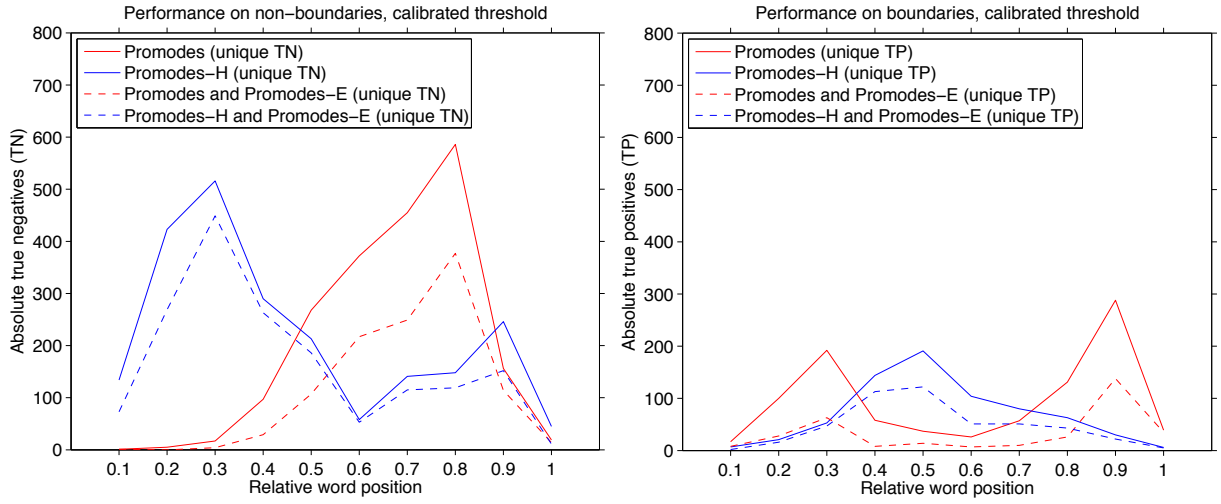
We would like to thank Narayanan Edakunni and Bruno Golénia for discussions concerning this paper as well as the anonymous reviewers for their comments. The research described was sponsored by EPSRC grant EP/E010857/1 *Learning the morphology of complex synthetic languages*.



(a) True negatives, default

(b) True positives, default

Figure 3: Analysis of results using default threshold.



(a) True negatives, calibrated

(b) True positives, calibrated

Figure 4: Analysis of results using calibrated threshold.

References

J. W. Amtrup. 2003. Morphology in machine translation systems: Efficient integration of finite state transducers and feature structure descriptions. *Machine Translation*, 18(3):217–238.

E. Atwell and A. Roberts. 2006. Combinatory hybrid elementary analysis of text (CHEAT). *Proceedings of the PASCAL Challenges Workshop on Unsupervised Segmentation of Words into Morphemes, Venice, Italy*.

M. Creutz. 2006. *Induction of the Morphology of Natural Language: Unsupervised Morpheme Segmentation with Application to Automatic Speech Recognition*. Ph.D. thesis, Helsinki University of Technology, Espoo, Finland.

J. Davis and M. Goadrich. 2006. The relationship between precision-recall and ROC curves. *International Conference on Machine Learning, Pittsburgh, PA*, 233–240.

M. Gasser. 1994. Modularity in a connectionist model of morphology acquisition. *Proceedings of the 15th conference on Computational linguistics*, 1:214–220.

J. Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27:153–198.

J. Goldsmith. 2009. *The Handbook of Computational Linguistics, chapter Segmentation and morphology*. Blackwell.

M. A. Hafer and S. F. Weiss. 1974. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10:371–385.

Z. S. Harris. 1955. From phoneme to morpheme. *Language*, 31(2):190–222.

T. Hirsimaki, M. Creutz, V. Siivola, M. Kurimo, S. Virpioja, and J. Pylkkonen. 2006. Unlimited vocabulary speech recognition with morph language models applied to Finnish. *Computer Speech And Language*, 20(4):515–541.

K. Kettunen. 2009. Reductive and generative approaches to management of morphological variation of keywords in monolingual information retrieval: An overview. *Journal of Documentation*, 65:267 – 290.

M. Kurimo, S. Virpioja, and V. T. Turunen. 2009. Overview and results of Morpho Challenge 2009. *Working notes for the CLEF 2009 Workshop, Corfu, Greece*.

C. Monson, K. Hollingshead, and B. Roark. 2009. Probabilistic ParaMor. *Working notes for the CLEF 2009 Workshop, Corfu, Greece*.

C. Monson. 2008. *ParaMor: From Paradigm Structure To Natural Language Morphology Induction*. Ph.D. thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA.

R. J. Mooney and M. E. Califf. 1996. Learning the past tense of English verbs using inductive logic programming. *Symbolic, Connectionist, and Statistical Approaches to Learning for Natural Language Processing*, 370–384.

S. Muggleton and M. Bain. 1999. Analogical prediction. *Inductive Logic Programming: 9th International Workshop, ILP-99, Bled, Slovenia*, 234.

K. Oflazer, S. Nirenburg, and M. McShane. 2001. Bootstrapping morphological analyzers by combining human elicitation and machine learning. *Computational. Linguistics*, 27(1):59–85.

D. Opitz and R. Maclin. 1999. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198.

D. E. Rumelhart and J. L. McClelland. 1986. *On learning the past tenses of English verbs*. MIT Press, Cambridge, MA, USA.

K. Shalonova, B. Golénia, and P. A. Flach. 2009. Towards learning morphology for under-resourced fusional and agglutinating languages. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(5):956965.

M. G. Snover and M. R. Brent. 2001. A Bayesian model for morpheme and paradigm identification. *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, 490 – 498.

M. G. Snover, G. E. Jarosz, and M. R. Brent. 2002. Unsupervised learning of morphology using a novel directed search algorithm: Taking the first step. *Proceedings of the ACL-02 workshop on Morphological and phonological learning*, 6:11–20.

S. Spiegler, B. Golénia, K. Shalonova, P. A. Flach, and R. Tucker. 2008. Learning the morphology of Zulu with different degrees of supervision. *IEEE Workshop on Spoken Language Technology*.

S. Spiegler, B. Golénia, and P. A. Flach. 2009. Promodes: A probabilistic generative model for word decomposition. *Working Notes for the CLEF 2009 Workshop, Corfu, Greece*.

S. Spiegler, B. Golénia, and P. A. Flach. 2010a. Unsupervised word decomposition with the Promodes algorithm. *In Multilingual Information Access Evaluation Vol. I, CLEF 2009, Corfu, Greece, Lecture Notes in Computer Science, Springer*.

S. Spiegler, A. v. d. Spuy, and P. A. Flach. 2010b. Ukwabelana - An open-source morphological Zulu corpus. *in review*.

R. Sproat. 1996. Multilingual text analysis for text-to-speech synthesis. *Nat. Lang. Eng.*, 2(4):369–380.