Semi-supervised extensions to Morfessor Baseline

Oskar Kohonen and **Sami Virpioja** and **Laura Leppänen** and **Krista Lagus**
Aalto University School of Science and Technology
Adaptive Informatics Research Centre
P.O. Box 15400, FI-00076 AALTO, Finland
firstname.lastname@tkk.fi

Abstract

We have extended Morfessor Baseline, which is a well-known method for unsupervised morphological segmentation, to semi-supervised learning. As submission to Morpho Challenge 2010, we provide results from three methods: The first one is based on the unsupervised algorithm, but includes a weight parameter that can be used to control the amount of segmentation. The second one applies the semi-supervised extension, where the labeled training data is used also during the learning. The third one is based on the second, but as an additional step we label the segments using a Hidden Markov Model trained on the labeled data.

1 Introduction

This work is based on Kohonen et al. (2010), where the Morfessor Baseline method (Creutz and Lagus, 2002; Creutz and Lagus, 2005; Creutz and Lagus, 2007) was extended to the semi-supervised case. Morfessor is a family of generative probabilistic models designed for modeling highly-inflecting and compounding languages (Creutz and Lagus, 2007). It induces a lexicon of word segments, called morphs, from the data. In the semi-supervised version, the training data contains labeled words with known gold standard segmentations. The lexicons that include those segments are favored if the words are added to the data likelihood function. In addition, a small set of word forms with gold standard analyzes can be used for tuning the respective weights of the annotated and unannotated data.

Kohonen et al. (2010) made also a simple experiment on labeling the segmentations provided by the Morfessor to the morpheme labels given in the training data. The results were encouraging considering the trivial labeling method. Here, we extend this line of research by training Hidden Markov Models (HMM) suitable for the task. This results not only in segmentation, but a full morphological analysis of the words.

2 Semi-supervised Morfessor Baseline

Let $\boldsymbol{\theta}$ be the parameters of the model, $\boldsymbol{D}_W$ be the set of word forms used for training the model and $\boldsymbol{D}_{W \to A}$ be a subset of words for which we know the correct morphs. Each word w_j in $\boldsymbol{D}_W$ has a corresponding variable Z_j that denotes how it is segmented. That is, its value is a sequence of morphs, $z_j = (m_{j1}, \ldots, m_{j|z_j|})$. The set of Z_j:s, $\boldsymbol{Z} = (Z_1, \ldots, Z_{|\boldsymbol{D}_W|})$ is a hidden variable that we want to estimate.

A generative model gives the joint distribution $P(W = w, Z = z \,|\, \boldsymbol{\theta})$ of words and their segmentations. Assuming that the sequence of morphs in z can produce only one word type, the probability is simply $P(Z = z \,|\, \boldsymbol{\theta})$ for that word, and zero otherwise. Instead of determining a *posteriori* probability distribution $P(\boldsymbol{\theta}|\boldsymbol{D}_W, \boldsymbol{D}_{W \to A})$ over model parameters $\boldsymbol{\theta}$ as in Bayesian modeling, we try to find a point estimate of $\boldsymbol{\theta}$ given a cost function to minimize. The three main aspects in this framework are:

- What is the family of the model, i.e., how probabilities $P(Z = z \,|\, \boldsymbol{\theta})$ and $P(\boldsymbol{\theta})$ are defined?
- What is the cost function to minimize for selecting $\boldsymbol{\theta}$?
- How to minimize the cost function, i.e., what is the training algorithm?

Next, we shortly describe the applied solution for each of them. Only the cost function differs from the unsupervised Morfessor Baseline.

2.1 Model family

The model family in Morfessor Baseline is relatively simple: The model parameters $\boldsymbol{\theta}$ encode a morph lexicon, which includes the properties of the morphs. Each morph m in the lexicon has a probability of occurring in a word, $P(M = m \mid \boldsymbol{\theta})$, and these probabilities are assumed to be independent.

During training, each word w_j is assumed to have only one possible analysis. Thus, instead of using the joint distribution $P(\boldsymbol{D}_W, \boldsymbol{Z} \mid \boldsymbol{\theta})$, we need to use the likelihood function only conditioned on the analyses of the observed words, $P(\boldsymbol{D}_W \mid \boldsymbol{Z}, \boldsymbol{\theta})$. The conditional likelihood is

$$\begin{aligned} & P(\boldsymbol{D}_W \mid \boldsymbol{Z} = \boldsymbol{z}, \boldsymbol{\theta}) \\ &= \prod_{j=1}^{|\boldsymbol{D}_W|} P(W = w_j \mid \boldsymbol{Z} = \boldsymbol{z}, \boldsymbol{\theta}) \\ &= \prod_{j=1}^{|\boldsymbol{D}_W|} \prod_{i=1}^{|z_j|} P(M = m_{ji} \mid \boldsymbol{\theta}), \end{aligned} \tag{1}$$

where m_{ij} is the i:th morph in word w_j.

The problem of using Equation 1 for the known segmentations in $\boldsymbol{D}_{W \to A}$ is that there can be alternative segmentations for each word. As a solution, we select only the segmentation that has the highest probability according to the model, and discard the others from the likelihood function. Due to practical reasons, the selection is done only after each training epoch (see Sec. 2.3).

The parameters $\boldsymbol{\theta}$ of the model are:

- Morph type count, or the size of the morph lexicon, $\mu \in \mathbb{Z}_+$
- Morph token count, or the number of morphs tokens in the observed data, $\nu \in \mathbb{Z}_+$
- Morph strings $(\sigma_1, \ldots, \sigma_\mu)$, $\sigma_i \in \Sigma^*$
- Morph counts $(\tau_1, \ldots, \tau_\mu)$, $\tau_i \in \{1, \ldots, \nu\}$, $\sum_i \tau_i = \nu$. Normalized with ν, these give the probabilities of the morphs.

In principle, each parameter has a prior probability. However, with MDL-inspired and non-informative priors, morph type count and morph token counts can be neglected as insignificant. The morph string prior is based on length distribution $P(L)$ and distribution $P(C)$ of characters over the character set Σ, both assumed to be known:

$$P(\sigma_i) = P(L = |\sigma_i|) \prod_{j=1}^{|\sigma_i|} P(C = \sigma_{ij}) \tag{2}$$

We applied the implicit length prior (Creutz and Lagus, 2005), where instead of determining $P(L)$, an end-of-word symbol is used as an additional character in $P(C)$. For morph counts, we used the non-informative prior

$$P(\tau_1, \ldots, \tau_\mu) = 1 / \binom{\nu - 1}{\mu - 1} \tag{3}$$

that gives equal probability to each possible combination of the counts when μ and ν are known.

2.2 Cost function

The unsupervised Morfessor algorithms try to find the maximum a posteriori estimate of the parameters. The equivalent cost function to minimize is

$$L(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{D}_W) = -\ln P(\boldsymbol{\theta}) - \ln P(\boldsymbol{D}_W \mid \boldsymbol{z}, \boldsymbol{\theta}). \tag{4}$$

In the semi-supervised version, we add the negative log-likelihood of the known segmentations in $\boldsymbol{D}_{W \to A}$. Furthermore, we weight the data likelihoods with parameters $\alpha > 0$ and $\beta > 0$:

$$\begin{aligned} L(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{D}_W, \boldsymbol{D}_{W \mapsto A}) = & \\ & - \ln P(\boldsymbol{\theta}) \\ & - \alpha \times \ln P(\boldsymbol{D}_W \mid \boldsymbol{z}, \boldsymbol{\theta}) \\ & - \beta \times \ln P(\boldsymbol{D}_{W \mapsto A} \mid \boldsymbol{z}, \boldsymbol{\theta}) \end{aligned} \tag{5}$$

The data likelihood weights control both the level of segmentation, as increasing the weight has to be compensated by a larger morph lexicon, and how large an effect the known segmentations have compared to the unsupervised segmentations.

2.3 Training algorithm

The training algorithm of Morfessor Baseline (Creutz and Lagus, 2005) tries to minimize the cost function by testing local changes to $\boldsymbol{z}$, modifying the parameters according to each change, and selecting the best one. The training algorithm is directly applicable to the semi-supervised case.

The initial parameters are obtained by adding all the words into the morph lexicon. Then, one word is processed at a time, and the segmentation that minimizes the cost function with the optimal

model parameters is selected and the parameters are updated respectively:

$$z_j^{(t+1)} = \arg\min_{z_j} \left\{ \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{z}^{(t)}, \boldsymbol{D}_W) \right\} \quad (6)$$

$$\boldsymbol{\theta}^{(t+1)} = \arg\min_{\boldsymbol{\theta}} \left\{ L(\boldsymbol{\theta}, \boldsymbol{z}^{(t+1)}, \boldsymbol{D}_W) \right\} \quad (7)$$

Because a probability of a morph does not depend on its context, the segmentations in $\boldsymbol{z}$ can be encoded as a tree-like graph, where the words are the top nodes and morphs the leaf nodes. In one training epoch, each top node is processed once. A node can either be left as it is or split into two parts. If the case of a split, the same test is applied recursively to its parts. As the changes cannot increase the cost function, the parameters will converge to a local optimum. In practice, the training is stopped when the average change in cost function per word in an epoch is smaller than 0.005.

3 Morpheme labeling

We use a first-order Hidden Markov Model (HMM) to label the induced morphs (segments of words) to morphemes. The unobserved states are the morpheme labels, and the observations are the segments. We construct the emission alphabet Σ by picking out all the morphs from both the training set and the segmented data that is to be labeled. The set of possible labels (states) is collected from the training data. When the training set does not provide labels for some morphs—as is the case for a large part of the morphs found in the Turkish training set—we group these morphs together under a separate label.

Labels of non-observable morphs, such as the plural morph in the word "men", are combined with the label of the preceding morph to create a compound label. In the case of the word "men" the compound label would be N+PL. Such compound labels are separated as post-processing. The resulting labeling would thus be "men_N +PL". Non-observable morphs that start a word are ignored altogether, since they are usually peculiarities in the gold standard labeling. For example, the English gold standard segmentation for the word "propjet" includes a non-observable prefix "turbo", which is clearly unnecessary.

Hyphens at the beginning or end of a morph such as the one in "-inspired", the second morph in a segmentation of the word "abba-inspired", are removed. I.e. "-inspired" is treated as the same morph as "inspired" without the hyphen. Hyphens that are segmented as morphs of their own are taken into account during the calculation of the Viterbi paths but are left out of the result files. Thus, the segmentation "educator - scientist" becomes "educator_N scientist_N" in the results.

Finally, we handle stem allomorphy by replacing morphs with their respective morphemes when provided by the training set. This is done as post-processing. For example, the segmentation "caricatur ish" becomes "caricature_N ish_s".

3.1 Transition and emission probabilities

After the sets of emissions and labels are collected, maximum likelihood estimation is applied to calculate state transition and emission probabilities from the training data. The probability of a transition from state l_1 to state l_2 is

$$P(l_2 \,|\, l_1) = \frac{C(l_1, l_2)}{C(l_1)}, \quad (8)$$

where $C(l_1, l_2)$ is the number of times l_2 follows l_1 in the training set and $C(l_2)$ is the total number of occurrences of l_2 in the training set.

Similarly, we can estimate that the probability that state l emits morph m is $C(m, l)/C(l)$, where $C(m, l)$ is the number of times m is tagged with l in the training set. However, to accommodate previously unseen morpheme emissions, we apply smoothing to emission probabilities. Smoothing is applied only for labels that represent open classes of morphs, that is, morph classes that can be expanded with new items. For Finnish and English these are nouns, verbs and adjectives. Because the gold standard does not provide labeling for Turkish nouns, verbs and adjectives, we have used the class of morphs that were unlabeled in the gold standard as the only open class when labeling the Turkish data.

As a smoothing method, we use absolute discounting. That is, we subtract a constant value $\delta = 0.1$ from all emission counts $C(m, l)$ greater than zero, and the remaining probability mass is then divided between the previously unseen emissions. Thus, if $N_0(l)$ is the number of emissions for label l with $C(m, l) = 0$, we get

$$P(m \,|\, l) = \begin{cases} \dfrac{C(m,l) - \delta}{C(l)} & \text{if } C(m,l) > 0 \\ \dfrac{(|\Sigma| - N_0(l))\delta}{N_0(l)C(l)} & \text{otherwise.} \end{cases} \quad (9)$$

4 Experiments

We compare four different variants of the Morfessor Baseline algorithm:

- **Unsupervised (U):** The classic, unsupervised Morfessor baseline.
- **Unsupervised + weighting (U+W):** A development set is used for adjusting the weight of the likelihood α. When $\alpha = 1$, the method is equivalent to the unsupervised baseline.
- **Semi-supervised + weighting (S+W):** The semi-supervised method trained with both annotated and unannotated data. The parameters α and β are optimized using the development set.
- **Semi-supervised + weighting + labeling (S+W+L):** As above, but the obtained morphs are labeled with a HMM tagger trained on the annotated training data.

All variants were trained for English, Finnish, and Turkish. Only the unsupervised models were trained for German, as there was no gold standard segmentations available for it. Only the data sets are from the Morpho Challenge 2010 web site[1] were applied. The provided development sets were used for optimizing α and β. The training sets of gold standard segmentations were used in training the semi-supervised segmentation models and the labeling models.

Table 1 shows the values for the optimal weights α and β that were chosen for different languages using the development set in both unsupervised and semi-supervised cases, as well as the respective results. The unsupervised method with weighting (U+W) results in more balanced precision and recall values than the unsupervised baseline method (U), thus clearly increasing the F-measures. The amount of increase is especially large for Finnish and Turkish languages due to the very low recall of the baseline.

The semi-supervised method (S+W) results in a considerable increase in recall and a somewhat more modest increase in precision for English and Finnish. For Turkish, however, we get the opposite result: a large improvement in precision and a small increase in recall. In both cases, the obtained F-measures are clearly better than the ones obtained with the unsupervised training.

[1] `www.cis.hut.fi/morphochallenge2010`

Model	α	β	P %	R %	F %
English					
U	-	-	84.75	44.28	58.17
U+W	0.25	-	67.32	60.73	63.86
S+W	0.5	1000	68.46	70.40	69.42
S+W+L	0.5	1000	73.05	68.12	70.50
Finnish					
U	-	-	84.48	17.45	28.92
U+W	0.01	-	59.26	47.00	52.42
S+W	0.01	2000	63.71	60.25	61.93
S+W+L	0.01	500	65.77	67.07	66.41
German					
U	-	-	70.85	22.32	33.95
U+W	0.05	-	56.40	48.53	52.17
Turkish					
U	-	-	94.18	16.85	28.58
U+W	0.01	-	44.91	47.10	45.98
S+W	0.1	1000	73.07	47.95	57.90
S+W+L	0.005	2500	76.95	60.59	67.80

Table 1: The optimal values for the weights α and β and the respective precision (P), recall (R) and F-measure (F) on the development set.

The morpheme labeling (L) should improve recall by solving allomorphy, i.e., finding common labels for the different surface forms, and precision by disambiguating surface forms of different morphemes. In our experiments, the largest increase in F-measure—nearly 10% absolute—is obtained for Turkish, for which the recall increases considerably. For Finnish, the increase is about 4.5% absolute. Both the Finnish and Turkish data sets include a large number of suffixes that have allomorphy, which explains the large improvements. English benefits less from the labeling, gaining slightly over 1% to the F-measure. The increase in precision is larger than for the other languages, but recall is, in fact, decreased by the labeling.

5 Conclusions

We have presented a semi-supervised extension to the Morfessor Baseline method, which performs morphological segmentation using maximum a posteriori estimation. Using gradually more of the information provided by the annotated data sets, we improve the F-measure results on the development set, e.g., for Finnish, from 29% to 52% by optimizing a weight parameter for the data likelihood, to 61% by using the annotated training data in the likelihood function, and finally to 66% by

using a Hidden Markov Model to label the segmentations. The method could be improved further by using the HMM probabilities directly when segmenting the words. The downside of our approach is that it requires word annotations with both the segmentations (morphs) and their labels (morphemes).

Acknowledgments

This work was funded by Academy of Finland, Graduate School of Language Technology in Finland, and the 7th Framework Programme of the European Commission through the META-NET project (249119).

References

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the Workshop on Morphological and Phonological Learning of ACL'02*, pages 21–30, Philadelphia, Pennsylvania, USA.

Mathias Creutz and Krista Lagus. 2005. Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. Technical Report A81, Publications in Computer and Information Science, Helsinki University of Technology.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1), January.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. Semi-supervised learning of concatenative morphology. In *Proceedings of the Eleventh Meeting of the ACL Special Interest Group on Computational Phonology and Morphology (SIGMORPHON 2010)*, pages 78–86, Uppsala, Sweden. Association for Computational Linguistics.