Supervised Morphological Segmentation in a Low-Resource Learning Setting using Conditional Random Fields

Teemu Ruokolainen[a] **Oskar Kohonen**[a] **Sami Virpioja**[a] **Mikko Kurimo**[b]
[a] Department of Information and Computer Science, Aalto University
[b] Department of Signal Processing and Acoustics, Aalto University
firstname.lastname@aalto.fi

Abstract

We discuss data-driven morphological segmentation, in which word forms are segmented into morphs, the surface forms of morphemes. Our focus is on a low-resource learning setting, in which only a small amount of annotated word forms are available for model training, while unannotated word forms are available in abundance. The current state-of-art methods 1) exploit both the annotated and unannotated data in a semi-supervised manner, and 2) learn morph lexicons and subsequently uncover segmentations by generating the most likely morph sequences. In contrast, we discuss 1) employing only the annotated data in a supervised manner, while entirely ignoring the unannotated data, and 2) directly learning to predict morph boundaries given their local sub-string contexts instead of learning the morph lexicons. Specifically, we employ conditional random fields, a popular discriminative log-linear model for segmentation. We present experiments on two data sets comprising five diverse languages. We show that the fully supervised boundary prediction approach outperforms the state-of-art semi-supervised morph lexicon approaches on all languages when using the same annotated data sets.

1 Introduction

Modern natural language processing (NLP) applications, such as speech recognition, information retrieval and machine translation, perform their tasks using statistical language models. For morphologically rich languages, estimation of the language models is problematic due to the high number of compound words and inflected word forms. A successful means of alleviating this data sparsity problem is to segment words into meaning-bearing sub-word units (Hirsimäki et al., 2006; Creutz et al., 2007; Turunen and Kurimo, 2011). In linguistics, the smallest meaning-bearing units of a language are called *morphemes* and their surface forms *morphs*. Thus, morphs are natural targets for the segmentation.

For most languages, existing resources contain large amounts of raw unannotated text data, only small amounts of manually prepared annotated training data, and no freely available rule-based morphological analyzers. The focus of our work is on performing morphological segmentation in this low-resource scenario. Given this setting, the current state-of-art methods approach the problem by learning *morph lexicons* from both annotated and unannotated data using *semi-supervised* machine learning techniques (Poon et al., 2009; Kohonen et al., 2010). Subsequent to model training, the methods uncover morph boundaries for new word forms by *generating* their most likely morph sequences according to the morph lexicons.

In contrast to learning morph lexicons (Poon et al., 2009; Kohonen et al., 2010), we study morphological segmentation by learning to directly predict *morph boundaries* based on their local substring contexts. Specifically, we apply the linear-chain conditional random field model, a popular *discriminative* log-linear model for segmentation presented originally by Lafferty et al. (2001). Importantly, we learn the segmentation model from solely the small annotated data in a *supervised* manner, while entirely ignoring the unannotated data. Despite not using the unannotated data, we show that by discriminatively learning to predict the morph boundaries, we are able to outperform the previous state-of-art.

We present experiments on Arabic and Hebrew using the data set presented originally by Snyder and Barzilay (2008), and on English, Finnish and

Turkish using the Morpho Challenge 2009/2010 data sets (Kurimo et al., 2009; Kurimo et al., 2010). The results are compared against two state-of-art techniques, namely the log-linear modeling approach presented by Poon et al. (2009) and the semi-supervised Morfessor algorithm (Kohonen et al., 2010). We show that when employing the same small amount of annotated training data, the CRF-based boundary prediction approach outperforms these reference methods on all languages. Additionally, since the CRF model learns from solely the small annotated data set, its training is computationally much less demanding compared to the semi-supervised methods, which utilize both the annotated and the unannotated data sets.

The rest of the paper is organized as follows. In Section 2, we discuss related work in morphological segmentation and methodology. In Section 3, we describe our segmentation method. Our experimental setup is described in Section 4, and the obtained results are presented in Section 5. In Section 6, we discuss the method and the results. Finally, we present conclusions on the work in Section 7.

2 Related work

The CRF model has been widely used in NLP segmentation tasks, such as shallow parsing (Sha and Pereira, 2003), named entity recognition (McCallum and Li, 2003), and word segmentation (Zhao et al., 2006). Recently, CRFs were also employed successfully in morphological segmentation for Arabic by Green and DeNero (2012) as a component of an English to Arabic machine translation system. While the segmentation method of Green and DeNero (2012) and ours is very similar, our focuses and contributions differ in several ways. First, while in our work we consider the low-resource learning setting, in which a small annotated data set is available (up to 3,130 word types), their model is trained on the Arabic Treebank (Maamouri et al., 2004) constituting several times larger training set (588,244 word tokens). Second, we present empirical comparison between the CRF approach and two state-of-art methods (Poon et al., 2009; Kohonen et al., 2010) on five diverse languages. Third, due to being a component of a larger system, their presentation on the method and experiments is rather underspeficied, while here we are able to provide a more thorough description.

In the experimental section, we compare the CRF-based segmentation approach with two state-of-art methods, the log-linear modeling approach presented by Poon et al. (2009) and the semi-supervised Morfessor algorithm (Kohonen et al., 2010). As stated previously, the CRF-based segmentation approach differs from these methods in that it learns to predict morph boundaries from a small amount of annotated data, in contrast to learning morph lexicons from both annotated and large amounts of unannotated data.

Lastly, there exists ample work on varying unsupervised (and semi-supervised) morphological segmentation methods. A useful review is given by Hammarström and Borin (2011). The fundamental difference between our approach and these techniques is that our method necessarily requires manually annotated training data.

3 Methods

In this section, we describe in detail the CRF-based approach for supervised morphological segmentation.

3.1 Morphological segmentation as a classification task

We represent the morphological segmentation task as a structured classification problem by assigning each character to one of four classes, namely {*beginning of a multi-character morph (B)*, *middle of a multi-character morph (M)*, *end of a multi-character morph (E)*, *single character morph (S)*}. For example, consider the English word form

drivers

with a corresponding segmentation

driv + er + s .

Using the classification notation, this segmentation is represented as

START	B	M	M	E	B	E	S	STOP
<w>	d	r	i	v	e	r	s	</w>

where we have assumed additional word start and end markers <*w*> and </*w*> with respective classes *START* and *STOP*. As another example, consider the Finnish word form

autoilla (with cars)

with a corresponding segmentation

auto + i + lla .

Using the classification notation, this segmentation is represented as

START	B	M	M	E	S	B	M	E	STOP
<w>	a	u	t	o	i	l	l	a	</w>

Intuitively, instead of the four class set {B, M, E, S}, a segmentation could be accomplished using only a set of two classes {B, M} as in (Green and DeNero, 2012). However, similarly to Chinese word segmentation (Zhao et al., 2006), our preliminary experiments suggested that using the more fine-grained four class set {B, M, E, S} performed slightly better. This result indicates that morph segments of differerent lengths behave differently.

3.2 Linear-chain conditional random fields

We perform the above structured classification using linear-chain conditional random fields (CRFs), a discriminative log-linear model for tagging and segmentation (Lafferty et al., 2001). The central idea of the linear-chain CRF is to exploit the dependencies between the output variables using a chain structured undirected graph, also referred to as a Markov random field, while conditioning the output globally on the observation.

Formally, the model for input $\boldsymbol{x}$ (characters in a word) and output $\boldsymbol{y}$ (classes corresponding to characters) is written as

$$p\left(\boldsymbol{y} \,|\, \boldsymbol{x}; \boldsymbol{w}\right) \propto \prod_{t=2}^{T} \exp\left(\boldsymbol{w}^{\top}\boldsymbol{f}(y_{t-1}, y_t, \boldsymbol{x}, t)\right), \tag{1}$$

where t indexes the characters, T denotes word length, $\boldsymbol{w}$ the model parameter vector, and $\boldsymbol{f}$ the vector-valued feature extracting function.

The purpose of the feature extraction function $\boldsymbol{f}$ is to capture the co-occurrence behavior of the tag transitions (y_{t-1}, y_t) and a set of features describing character position t of word form $\boldsymbol{x}$. The strength of the CRF model lies in its capability to utilize arbitrary, non-independent features.

3.3 Feature extraction

The quality of the segmentation depends heavily on the choice of features defined by the feature extraction function $\boldsymbol{f}$. We will next describe and motivate the feature set used in the experiments.

Our feature set consists of binary indicator functions describing the position t of word $\mathbf{x}$ using all left and right substrings up to a maximum length δ. For example, consider the problem of deciding if the letter e in the word *drivers* is preceded by a morph boundary. This decision is now based on the overlapping substrings to the left and right of this potential boundary position, that is {*v*, *iv*, *riv*, *driv*, *<w>driv*} and {*e*, *er*, *ers*, *ers</w>*}, respectively. The substrings to the left and right are considered independently. Naturally, if the maximum allowed substring length δ is less than five, the longest substrings are discarded accordingly. In general, the optimum δ depends on both the amount of available training data and the language.

In addition to the substring functions, we use a bias function which returns value 1 independent of the input $\boldsymbol{x}$. The bias and substring features are combined with all the possible tag transitions.

To motivate this choice of feature set, consider formulating an intuitive segmentation rule for the English words *talked*, *played* and *speed* with the correct segmentations *talk* + *ed*, *play* + *ed* and *speed*, respectively. Now, as a right context *ed* is generally a strong indicator of a boundary, one could first formulate a rule

position t is a segment boundary
if its right context is *ed*.

This rule would indeed correctly segment the words *talked* and *played*, but would incorrectly segment *speed* as *spe* + *ed*. This error can be resolved if the left contexts are utilized as inhibitors by expanding the above rule as

position t is a segment boundary
if its right context is *ed*
and the left context is not *spe*.

Using the feature set defined above, the CRF model can learn to perform segmentation in this rule-like manner according to the training data. For example, using the above example words and segmentations for training, the CRFs could learn to assign a high score for a boundary given that the right context is *ed* and a high score for a non-boundary given the left context *spe*. Subsequent to training, making segmentation decisions for new word forms can then be interpreted as voting based on these scores.

3.4 Parameter estimation

The CRF model parameters $\boldsymbol{w}$ are estimated based on an annotated training data set. Common training criteria include the maximum likelihood (Lafferty et al., 2001; Peng et al., 2004; Zhao et al., 2006), averaged structured perceptron (Collins, 2002), and max-margin (Szummer et al., 2008). In this work, we estimate the parameters using the perceptron algorithm (Collins, 2002).

In perceptron training, the required graph inference can be efficiently performed using the standard Viterbi algorithm. Subsequent to training, the segmentations for test instances are acquired again using Viterbi search.

Compared to other training criteria, the structured perceptron has the advantage of employing only a single hyperparameter, namely the number of passes over training data, making model estimation fast and straightforward. We optimize the hyperparameter using a separate development set. Lastly, we consider the longest substring length δ a second hyperparameter optimized using the development set.

4 Experimental setup

This section describes the data sets, evaluation metrics, reference methods, and other details concerning the evaluation of the methods.

4.1 Data sets

We evaluate the methods on two different data sets comprising five languages in total.

S&B data. The first data set we use is the Hebrew Bible parallel corpus introduced by Snyder and Barzilay (2008). It contains 6,192 parallel phrases in Hebrew, Arabic, Aramaic, and English and their frequencies (ranging from 5 to 3517). The phrases have been extracted using automatic word alignment. The Hebrew and Arabic phrases have manually annotated morphological segmentations, and they are used in our experiments. The phrases are sorted according to frequency, and every fifth phrase starting from the first phrase is placed in the test set, every fifth starting from the second phrase in the development set (up to 500 phrases), and the rest of the phrases in the training set. [1] The total numbers of word types in the sets are shown in Table 1. Finally, the word forms in the training set are randomly permuted, and the first 25%, 50%, 75%, and 100% of them are selected as subsets to study the effect of training data size.

MC data. The second data set is based on the Morpho Challenge 2010 (Kurimo et al., 2010). It includes manually prepared morphological segmentations in English, Finnish and Turkish. The additional German corpus does not have segmentation annotation and is therefore excluded. The annotated data sets include training, development, and test sets for each language. Following Virpioja et al. (2011), the test set results are based on ten randomly selected 1,000 word sets. Moreover, we divide the annotated training sets into ten partitions with respective sizes of 100, 200, . . . , 1000 words so that each partition is a subset of the all larger partitions. The data is divided so that the smallest set had every 10th word of the original set, the second set every 10th word and the following word, and so forth. For reference methods that require unannotated data, we use the English, Finnish and Turkish corpora from Competition 1 of Morpho Challenge 2009 (Kurimo et al., 2009). Table 2 shows the sizes of the MC data sets.

	Arabic	Hebrew
Training	3,130	2,770
Development	472	450
Test	1,107	1,040

Table 1: The numbers of word types in S&B data sets (Snyder and Barzilay, 2008).

	English	Finnish	Turkish
Unannot.	384,903	2,206,719	617,298
Training	1,000	1,000	1,000
Develop.	694	835	763
Test	10×1,000	10×1,000	10×1,000

Table 2: The numbers of word types in the MC data sets (Kurimo et al., 2009; Kurimo et al., 2010).

4.2 Evaluation measures

The word segmentations are evaluated by comparison with linguistic morphs using *precision*, *recall*, and *F-measure*. The F-measure equals the geometric mean of precision (the percentage of correctly assigned boundaries with respect to all assigned boundaries) and recall (the percentage of correctly assigned boundaries with respect to the reference boundaries). While using F-measure is a standard procedure, the prior work differ at least in three details: (1) whether precision and recall are calculated as *micro-average* over all segmentation points or as *macro-average* over all the word forms, (2) whether the evaluation is based on word *types* or word *tokens* in a corpus, and (3) if the

[1] We are grateful to Dr. Hoifung Poon for providing us instructions for dividing of the data set.

reference segmentations have *alternative* correct choices for a single word type, and how to deal with them.

For the experiments with the S&B data sets, we follow Poon et al. (2009) and apply token-based micro-averages. For the experiments with the MC data sets, we follow Virpioja et al. (2011) and use type-based macro-averages. However, differing from their boundary measure, we take the best match over the alternative reference analyses (separately for precision and recall), since none of the methods considered here provide multiple segmentations per word type. For the models trained with the full training set, we also report the F-measures of the boundary evaluation method by Virpioja et al. (2011) in order to compare to the results reported in the Morpho Challenge website.

4.3 CRF feature extraction and training

The features included in the feature vector in the CRF model (1) are described in Section 3.3. We include all substring features which occur in the training data.

The CRF model is trained using the averaged perceptron algorithm as described in Section 3.4. The algorithm initializes the model parameters with zero vectors. The model performance, measured using F-measure, is evaluated on the development set after each pass over the training set, and the training is terminated when the performance has not improved during last 5 passes. The maximum length of substrings δ is optimized by considering $\delta = 1, 2, 3, \ldots$, and the search is terminated when the performance has not improved during last 5 values. Finally, the algorithm returns the parameters yielding the highest F-measure on the development set.

For some words, the MC training sets include several alternative segmentations. We resolve this ambiguity by using the first given alternative and discarding the rest. During evaluation, the alternative segmentations are taken into account as described in Section 4.2.

The experiments are run on a standard desktop computer using our own single-threaded Python-based implementation[2].

4.4 Reference methods

We compare our method's performance on Arabic and Hebrew data with semi-supervised Morfessor (Kohonen et al., 2010) and the results reported by Poon et al. (2009). On Finnish, English and Turkish data, we compare the method only with semi-supervised Morfessor as we have no implementation of the model by Poon et al. (2009).

We use a recently released Python implementation of semi-supervised Morfessor[3]. Semi-supervised Morfessor was trained separately for each training set size, always using the full unannotated data sets in addition to the annotated sets. The hyperparameters, the unannotated data weight α and the annotated data weight β, were optimized with a grid search on the development set. For the S&B data, there are no separate unannotated sets. When the annotated training set size is varied, the remaining parts are utilized as unannotated data.

The log-linear model described in (Poon et al., 2009) and the semi-supervised Morfessor algorithm are later referred to as POON-2009 and S-MORFESSOR for brevity.

5 Results

Method performances for Arabic and Hebrew on the S&B data are presented in Tables 3 and 4, respectively. The results for the POON-2009 model are extracted from (Poon et al., 2009). Performances for English, Finnish and Turkish on the MC data set are presented in Tables 5, 6 and 7, respectively.

On the Arabic and Hebrew data sets, the CRFs outperform POON-2009 and S-MORFESSOR substantially on all the considered data set sizes. On Finnish and Turkish data, the CRFs outperform S-MORFESSOR except for the smallest sets of 100 instances. On English data, the CRFs outperform S-MORFESSOR when the training set is 500 instances or larger.

Using our implementation of the CRF model, obtaining the results for Arabic, Hebrew, English, Finnish, and Turkish consumed 10, 11, 22, 32, and 28 minutes, respectively. These CPU times include model training and hyperparameter optimization. In comparison, S-MORFESSOR training is considerably slower. For Arabic and Hebrew, the S-MORFESSOR total training times were 24 and 22 minutes, respectively, and for English, Finnish, and Turkish 4, 22, and 10 days, respectively. The higher training times of S-MORFESSOR are partly because of the larger

[2] Available at `http://users.ics.aalto.fi/tpruokol/`

[3] Available at `https://github.com/aalto-speech/morfessor`

grids in hyperparameter optimization. Furthermore, the S-MORFESSOR training time for each grid point grows linearly with the size of the unannotated data set, resulting in particularly slow training on the MC data sets. All reported times are total CPU times for single-threaded runs, while in practice grid searches can be parallelized.

The perceptron algorithm typically converged after 10 passes over the training set, and never required more than 40 passes to terminate. Depending on the size of the training data, the optimized maximum lengths of substrings varied in ranges {3,5}, {2,7}, {3,9}, {3,6}, {3,7}, for Arabic, Hebrew, English, Finnish and Turkish, respectively.

Method	%Lbl.	Prec.	Rec.	F1
CRF	25	95.5	93.1	**94.3**
S-MORFESSOR	25	78.7	79.7	79.2
POON-2009	25	84.9	85.5	85.2
CRF	50	96.5	94.6	**95.5**
S-MORFESSOR	50	87.5	91.5	89.4
POON-2009	50	88.2	86.2	87.5
CRF	75	97.2	96.1	**96.6**
S-MORFESSOR	75	92.8	83.0	87.6
POON-2009	75	89.6	86.4	87.9
CRF	100	98.1	97.5	**97.8**
S-MORFESSOR	100	91.4	91.8	91.6
POON-2009	100	91.7	88.5	90.0

Table 3: Results for Arabic on the S&B data set (Snyder and Barzilay, 2008). The column titled *%Lbl.* denotes the percentage of the annotated data used for training. In addition to the given percentages of annotated data, POON-2009 and S-MORFESSOR utilized the remainder of the data as an unannotated set.

Finally, Table 8 shows the results of the CRF and S-MORFESSOR models trained with the full English, Finnish, and Turkish MC data sets and evaluated with the boundary evaluation method of Virpioja et al. (2011). That is, these numbers are directly comparable to the BPR-F column in the result tables presented at the Morpho Challenge website[4]. For each of the three languages, CRF clearly outperforms all the Morpho Challenge submissions that have provided morphological segmentations.

Method	%Lbl.	Prec.	Rec.	F1
CRF	25	90.5	90.6	**90.6**
S-MORFESSOR	25	71.5	85.3	77.8
POON-2009	25	78.7	73.3	75.9
CRF	50	94.0	91.5	**92.7**
S-MORFESSOR	50	82.1	81.8	81.9
POON-2009	50	82.8	74.6	78.4
CRF	75	94.0	92.7	**93.4**
S-MORFESSOR	75	84.0	88.1	86.0
POON-2009	75	83.1	77.3	80.1
CRF	100	94.9	94.0	**94.5**
S-MORFESSOR	100	85.3	91.1	88.1
POON-2009	100	83.0	78.9	80.9

Table 4: Results for Hebrew on the S&B data set (Snyder and Barzilay, 2008). The column titled *%Lbl.* denotes the percentage of the annotated data used for training. In addition to the given percentages of annotated data, POON-2009 and S-MORFESSOR utilized the remainder of the data as an unannotated set.

6 Discussion

Intuitively, the CRF-based supervised learning approach should yield high segmentation accuracy when there are large amounts of annotated training data available. However, perhaps surprisingly, the CRF model yields state-of-art results already using very small amounts of training data. This result is meaningful since for most languages it is infeasible to acquire large amounts of annotated training data.

The strength of the discriminatively trained CRF model is that overlapping, non-independent features can be naturally employed. Importantly, we showed that simple, language-independent substring features are sufficient for high performance. However, adding new, task- and language-dependent features is also easy. One might, for example, explore features capturing vowel harmony in Finnish and Turkish.

The CRFs was estimated using the structured perceptron algorithm (Collins, 2002), which has the benefit of being computationally efficient and easy to implement. Other training criteria, such as maximum likelihood (Lafferty et al., 2001) or max-margin (Szummer et al., 2008), could also be employed. Similarly, other classifiers, such as the Maximum Entropy Markov Models (MEMMs) (McCallum et al., 2000), are applicable. However, as the amount of information in-

[4] `http://research.ics.aalto.fi/events/morphochallenge/`

Method	Train.	Prec.	Rec.	F1
CRF	100	80.2	74.6	77.3
S-MORFESSOR	100	88.1	79.7	**83.7**
CRF	200	84.7	79.2	81.8
S-MORFESSOR	200	88.1	79.5	**83.6**
CRF	300	86.7	79.8	83.1
S-MORFESSOR	300	88.4	80.6	**84.3**
CRF	400	86.5	80.6	83.4
S-MORFESSOR	400	84.6	83.6	**84.1**
CRF	500	88.6	80.7	**84.5**
S-MORFESSOR	500	86.3	82.7	84.4
CRF	600	88.1	82.6	**85.3**
S-MORFESSOR	600	86.7	82.5	84.5
CRF	700	87.9	83.4	**85.6**
S-MORFESSOR	700	86.0	82.9	84.4
CRF	800	89.1	83.2	**86.1**
S-MORFESSOR	800	87.1	82.5	84.8
CRF	900	89.0	82.9	**85.8**
S-MORFESSOR	900	86.4	82.6	84.5
CRF	1000	89.8	83.5	**86.5**
S-MORFESSOR	1000	88.8	80.1	84.3

Table 5: Results for English on the Morpho Challenge 2009/2010 data set (Kurimo et al., 2009; Kurimo et al., 2010). The column titled *Train.* denotes the number of annotated training instances. In addition to the annotated data, S-MORFESSOR utilized an unannotated set of 384,903 word types.

Method	Train.	Prec.	Rec.	F1
CRF	100	71.4	66.0	68.6
S-MORFESSOR	100	69.8	71.0	**70.4**
CRF	200	76.4	71.3	**73.8**
S-MORFESSOR	200	75.5	68.6	71.9
CRF	300	80.4	73.9	**77.0**
S-MORFESSOR	300	73.1	71.8	72.5
CRF	400	81.0	76.6	**78.7**
S-MORFESSOR	400	73.3	74.3	73.8
CRF	500	82.9	77.9	**80.3**
S-MORFESSOR	500	73.5	75.1	74.3
CRF	600	82.6	80.6	**81.6**
S-MORFESSOR	600	76.1	73.7	74.9
CRF	700	84.3	81.4	**82.8**
S-MORFESSOR	700	75.0	76.6	75.8
CRF	800	85.1	83.4	**84.2**
S-MORFESSOR	800	74.1	78.2	76.1
CRF	900	85.2	83.8	**84.5**
S-MORFESSOR	900	74.2	78.5	76.3
CRF	1000	86.0	84.7	**85.3**
S-MORFESSOR	1000	74.2	78.8	76.4

Table 6: Results for Finnish on the Morpho Challenge 2009/2010 data set (Kurimo et al., 2009; Kurimo et al., 2010). The column titled *Train.* denotes the number of annotated training instances. In addition to the annotated data, S-MORFESSOR utilized an unannotated set of 2,206,719 word types.

corporated in the model would be unchanged, the choice of parameter estimation criterion and classifier is unlikely to have a dramatic effect on the method performance.

In CRF training, we focused on the supervised learning scenario, in which no unannotated data is exploited in addition to the annotated training sets. However, there does exist ample work on extending CRF training to the semi-supervised setting (for example, see Mann and McCallum (2008) and the references therein). Nevertheless, our results strongly suggest that it is crucial to use the few available annotated training instances as efficiently as possible before turning model training burdensome by incorporating large amounts of unannotated data.

Following previous work (Poon et al., 2009; Kohonen et al., 2010; Virpioja et al., 2011), we applied the boundary F-score evaluation measure, while Green and DeNero (2012) reported character accuracy. We consider the boundary F-score a better measure than accuracy, since the boundary-tag distribution is strongly skewed towards non-boundaries. Nevertheless, for completeness, we computed the character accuracy for our Arabic data set, obtaining the accuracy 99.1%, which is close to their reported accuracy of 98.6%. However, these values are not directly comparable due to our use of the Bible corpus by Snyder and Barzilay (2008) and their use of the Penn Arabic Treebank (Maamouri et al., 2004).

7 Conclusions

We have presented an empirical study in data-driven morphological segmentation employing supervised boundary prediction methodology. Specifically, we applied conditional random fields, a discriminative log-linear model for segmentation and tagging. From a methodological perspective, this approach differs from the previous state-of-art methods in two fundamental aspects. First, we utilize a discriminative model estimated using only annotated data. Second, we learn to predict morph

Method	Train.	Prec.	Rec.	F1
CRF	100	72.4	79.6	75.8
S-MORFESSOR	100	77.9	78.5	**78.2**
CRF	200	83.2	82.3	**82.8**
S-MORFESSOR	200	80.0	83.2	81.6
CRF	300	83.9	85.9	**84.9**
S-MORFESSOR	300	80.1	85.6	82.8
CRF	400	86.4	86.5	**86.4**
S-MORFESSOR	400	80.7	87.1	83.8
CRF	500	87.5	86.4	**87.0**
S-MORFESSOR	500	81.0	87.2	84.0
CRF	600	87.8	88.1	**87.9**
S-MORFESSOR	600	80.5	89.9	85.0
CRF	700	89.1	88.3	**88.7**
S-MORFESSOR	700	80.9	90.7	85.5
CRF	800	88.6	90.3	**89.4**
S-MORFESSOR	800	81.2	91.0	85.9
CRF	900	89.2	89.8	**89.5**
S-MORFESSOR	900	81.4	91.2	86.0
CRF	1000	89.9	90.4	**90.2**
S-MORFESSOR	1000	83.0	91.5	87.0

Table 7: Results for Turkish on the Morpho Challenge 2009/2010 data set (Kurimo et al., 2009; Kurimo et al., 2010). The column titled *Train.* denotes the number of annotated training instances. In addition to the annotated data, S-MORFESSOR utilized an unannotated set of 617,298 word types.

boundaries based on their local character substring contexts instead of learning a morph lexicon.

We showed that our supervised method yields improved results compared to previous state-of-art semi-supervised methods using the same small amount of annotated data, while not utilizing the unannotated data used by the reference methods. This result has two implications. First, supervised methods can provide excellent results in morphological segmentation already when there are only a few annotated training instances available. This is meaningful since for most languages it is infeasible to acquire large amounts of annotated training data. Second, performing morphological segmentation by directly modeling segment boundaries can be advantageous compared to modeling morph lexicons.

A potential direction for future work includes evaluating the morphs obtained by our method in real world applications, such as speech recognition and information retrieval. We are also interested in extending the method from fully supervised to semi-supervised learning.

Method	English	Finnish	Turkish
CRF	**82.0**	**81.9**	**71.5**
S-MORFESSOR	79.6	73.5	70.5

Table 8: F-measures of the Morpho Challenge boundary evaluation for CRF and S-MORFESSOR using the full annotated training data set.

Acknowledgements

This work was financially supported by Langnet (Finnish doctoral programme in language studies) and the Academy of Finland under the Finnish Centre of Excellence Program 2012–2017 (grant no. 251170), project *Multimodally grounded language technology* (no. 254104), and LASTU Programme (nos. 256887 and 259934).

References

M. Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, volume 10, pages 1–8. Association for Computational Linguistics.

M. Creutz, T. Hirsimäki, M. Kurimo, A. Puurula, J. Pylkkönen, V. Siivola, M. Varjokallio, E. Arisoy, M. Saraçlar, and A Stolcke. 2007. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Transactions on Speech and Language Processing*, 5(1):3:1–3:29, December.

S. Green and J. DeNero. 2012. A class-based agreement model for generating accurately inflected translations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 146–155. Association for Computational Linguistics.

H. Hammarström and L. Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350, June.

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

O. Kohonen, S. Virpioja, and K. Lagus. 2010. Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology* *and Phonology*, pages 78–86, Uppsala, Sweden, July. Association for Computational Linguistics.

M. Kurimo, S. Virpioja, V. Turunen, G. W. Blackwood, and W. Byrne. 2009. Overview and results of Morpho Challenge 2009. In *Working Notes for the CLEF 2009 Workshop*, Corfu, Greece, September.

M. Kurimo, S. Virpioja, and V. Turunen. 2010. Overview and results of Morpho Challenge 2010. In *Proceedings of the Morpho Challenge 2010 Workshop*, pages 7–24, Espoo, Finland, September. Aalto University School of Science and Technology, Department of Information and Computer Science. Technical Report TKK-ICS-R37.

J. Lafferty, A. McCallum, and F.C.N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki. 2004. The penn arabic treebank: Building a large-scale annotated arabic corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109.

G. Mann and A. McCallum. 2008. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proceedings of ACL-08: HLT*, pages 870–878. Association for Computational Linguistics.

A. McCallum and W. Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics.

A. McCallum, D. Freitag, and F. Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, pages 591–598, Stanford, CA, USA. Morgan Kaufmann.

F. Peng, F. Feng, and A. McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, page 562. Association for Computational Linguistics.

H. Poon, C. Cherry, and K. Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 209–217. Association for Computational Linguistics.

F. Sha and F. Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 134–141. Association for Computational Linguistics.

B. Snyder and R. Barzilay. 2008. Crosslingual propagation for morphological analysis. In *Proceedings of the AAAI*, pages 848–854.

M. Szummer, P. Kohli, and D. Hoiem. 2008. Learning CRFs using graph cuts. *Computer Vision–ECCV 2008*, pages 582–595.

V. Turunen and M. Kurimo. 2011. Speech retrieval from unsegmented Finnish audio using statistical morpheme-like units for segmentation, recognition, and retrieval. *ACM Transactions on Speech and Language Processing*, 8(1):1:1–1:25, October.

S. Virpioja, V. Turunen, S. Spiegler, O. Kohonen, and M. Kurimo. 2011. Empirical comparison of evaluation methods for unsupervised learning of morphology. *Traitement Automatique des Langues*, 52(2):45–90.

H. Zhao, C.N. Huang, and M. Li. 2006. An improved chinese word segmentation system with conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, volume 1082117. Sydney: July.