Unsupervised Morphological Segmentation with Log-Linear Models

Hoifung Poon*
Dept. of Computer Sci. & Eng.
University of Washington
Seattle, WA 98195
hoifung@cs.washington.edu

Colin Cherry
Microsoft Research
Redmond, WA 98052
colinc@microsoft.com

Kristina Toutanova
Microsoft Research
Redmond, WA 98052
kristout@microsoft.com

Abstract

Morphological segmentation breaks words into morphemes (the basic semantic units). It is a key component for natural language processing systems. Unsupervised morphological segmentation is attractive, because in every language there are virtually unlimited supplies of text, but very few labeled resources. However, most existing model-based systems for unsupervised morphological segmentation use directed generative models, making it difficult to leverage arbitrary overlapping features that are potentially helpful to learning. In this paper, we present the first log-linear model for unsupervised morphological segmentation. Our model uses overlapping features such as morphemes and their contexts, and incorporates exponential priors inspired by the minimum description length (MDL) principle. We present efficient algorithms for learning and inference by combining contrastive estimation with sampling. Our system, based on monolingual features only, outperforms a state-of-the-art system by a large margin, even when the latter uses bilingual information such as phrasal alignment and phonetic correspondence. On the Arabic Penn Treebank, our system reduces F1 error by 11% compared to Morfessor.

1 Introduction

The goal of morphological segmentation is to segment words into *morphemes*, the basic syntactic/semantic units. This is a key subtask in many NLP applications, including machine translation, speech recognition and question answering. Past approaches include rule-based morphological analyzers (Buckwalter, 2004) and supervised learning (Habash and Rambow, 2005). While successful, these require deep language expertise and a long and laborious process in system building or labeling.

Unsupervised approaches are attractive due to the the availability of large quantities of unlabeled text, and unsupervised morphological segmentation has been extensively studied for a number of languages (Brent et al., 1995; Goldsmith, 2001; Dasgupta and Ng, 2007; Creutz and Lagus, 2007). The lack of supervised labels makes it even more important to leverage rich features and global dependencies. However, existing systems use directed generative models (Creutz and Lagus, 2007; Snyder and Barzilay, 2008b), making it difficult to extend them with arbitrary overlapping dependencies that are potentially helpful to segmentation.

In this paper, we present the first log-linear model for unsupervised morphological segmentation. Our model incorporates simple priors inspired by the minimum description length (MDL) principle, as well as overlapping features such as morphemes and their contexts (e.g., in Arabic, the string *Al* is likely a morpheme, as is any string between *Al* and a word boundary). We develop efficient learning and inference algorithms using a novel combination of two ideas from previous work on unsupervised learning with log-linear models: contrastive estimation (Smith and Eisner, 2005) and sampling (Poon and Domingos, 2008).

We focus on inflectional morphology and test our

* This research was conducted during the author's internship at Microsoft Research.

approach on datasets in Arabic and Hebrew. Our system, using monolingual features only, outperforms Snyder & Barzilay (2008b) by a large margin, even when their system uses bilingual information such as phrasal alignment and phonetic correspondence. On the Arabic Penn Treebank, our system reduces F1 error by 11% compared to Morfessor Categories-MAP (Creutz and Lagus, 2007). Our system can be readily applied to supervised and semi-supervised learning. Using a fraction of the labeled data, it already outperforms Snyder & Barzilay's supervised results (2008a), which further demonstrates the benefit of using a log-linear model.

2 Related Work

There is a large body of work on the unsupervised learning of morphology. In addition to morphological segmentation, there has been work on unsupervised morpheme analysis, where one needs to determine features of word forms (Kurimo et al., 2007) or identify words with the same lemma by modeling stem changes (Schone and Jurafsky, 2001; Goldsmith, 2001). However, we focus our review specifically on morphological segmentation.

In the absence of labels, unsupervised learning must incorporate a strong learning bias that reflects prior knowledge about the task. In morphological segmentation, an often-used bias is the minimum description length (MDL) principle, which favors compact representations of the lexicon and corpus (Brent et al., 1995; Goldsmith, 2001; Creutz and Lagus, 2007). Other approaches use statistics on morpheme context, such as conditional entropy between adjacent n-grams, to identify morpheme candidates (Harris, 1955; Keshava and Pitler, 2006). In this paper, we incorporate both intuitions into a simple yet powerful model, and show that each contributes significantly to performance.

Unsupervised morphological segmentation systems also differ from the engineering perspective. Some adopt a pipeline approach (Schone and Jurafsky, 2001; Dasgupta and Ng, 2007; Demberg, 2007), which works by first extracting candidate affixes and stems, and then segmenting the words based on the candidates. Others model segmentation using a joint probabilistic distribution (Goldwater et al., 2006; Creutz and Lagus, 2007; Snyder and Barzilay, 2008b); they learn the model parameters from unlabeled data and produce the most probable segmentation as the final output. The latter approach is arguably more appealing from the modeling standpoint and avoids error propagation along the pipeline. However, most existing systems use directed generative models; Creutz & Lagus (2007) used an HMM, while Goldwater et al. (2006) and Snyder & Barzilay (2008b) used Bayesian models based on Pitman-Yor or Dirichlet processes. These models are difficult to extend with arbitrary overlapping features that can help improve accuracy.

In this work we incorporate novel overlapping contextual features and show that they greatly improve performance. Non-overlapping contextual features previously have been used in directed generative models (in the form of Markov models) for unsupervised morphological segmentation (Creutz and Lagus, 2007) or word segmentation (Goldwater et al., 2007). In terms of feature sets, our model is most closely related to the constituent-context model proposed by Klein and Manning (2001) for grammar induction. If we exclude the priors, our model can also be seen as a semi-Markov conditional random field (CRF) model (Sarawagi and Cohen, 2004). Semi-Markov CRFs previously have been used for supervised word segmentation (Andrew, 2006), but not for unsupervised morphological segmentation.

Unsupervised learning with log-linear models has received little attention in the past. Two notable exceptions are Smith & Eisner (2005) for POS tagging, and Poon & Domingos (2008) for coreference resolution. Learning with log-linear models requires computing the normalization constant (a.k.a. the partition function) Z. This is already challenging in supervised learning. In unsupervised learning, the difficulty is further compounded by the absence of supervised labels. Smith & Eisner (2005) proposed *contrastive estimation*, which uses a small neighborhood to compute Z. The neighborhood is carefully designed so that it not only makes computation easier but also offers sufficient contrastive information to aid unsupervised learning. Poon & Domingos (2008), on the other hand, used sampling to approximate Z.[1] In this work, we benefit from both techniques: contrastive estimation creates a manageable,

[1] Rosenfeld (1997) also did this for language modeling.

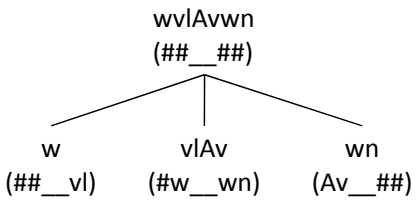

Figure 1: The morpheme and context (in parentheses) features for the segmented word *w-vlAv-wn*.

informative Z, while sampling enables the use of powerful global features.

3 Log-Linear Model for Unsupervised Morphological Segmentation

Central to our approach is a log-linear model that defines the joint probability distribution for a corpus (i.e., the words) and a segmentation on the corpus. The core of this model is a morpheme-context model, with one feature for each morpheme,[2] and one feature for each morpheme context. We represent contexts using the n-grams before and after the morpheme, for some constant n. To illustrate this, a segmented Arabic corpus is shown below along with its features, assuming we are tracking bigram contexts. The segmentation is indicated with hyphens, while the hash symbol (#) represents the word boundary.

Segmented Corpus hnAk w-vlAv-wn bn-w Al-ywm Al-jmAEp

Morpheme Feature:Value hnAk:1 w:2 vlAv:1 wn:1 bn:1 Al:2 ywm:1 jmAEp:1 hnAk:1 wvlAvwn:1 bnw:1 Alywm:1 Aljmaep:1

Bigram Context Feature:Value ##_vl:1 #w_wn:1 Av_##:1 ##_w#:1 bn_##:1 ##_yw:1 Al_##:2 ##_jm:1 ##_##:5

Furthermore, the corresponding features for the segmented word *w-vlAv-wn* are shown in Figure 1.

Each feature is associated with a weight, which correlates with the likelihood that the corresponding morpheme or context marks a valid morphological segment. Such overlapping features allow us to capture rich segmentation regularities. For example, given the Arabic word *Alywm*, to derive its correct segmentation *Al-ywm*, it helps to know that *Al* and *ywm* are likely morphemes whereas *Aly* or *lyw* are not; it also helps to know that *Al_##* or *##_yw* are likely morpheme contexts whereas *ly_##* or *##_wm* are not. Ablation tests verify the importance of these overlapping features (see Section 7.2).

Our morpheme-context model is inspired by the constituent-context model (CCM) proposed by Klein and Manning (2001) for grammar induction. The morphological segmentation of a word can be viewed as a flat tree, where the root node corresponds to the word and the leaves correspond to morphemes (see Figure 1). The CCM uses unigrams for context features. For this task, however, we found that bigrams and trigrams lead to much better accuracy. We use trigrams in our full model.

For learning, one can either view the corpus as a collection of word *types* (unique words) or *tokens* (word occurrences). Some systems (e.g., Morfessor) use token frequency for parameter estimation. Our system, however, performs much better using word types. This has also been observed for other morphological learners (Goldwater et al., 2006). Thus we use types in learning and inference, and effectively enforce the constraint that words can have only one segmentation per type. Evaluation is still based on tokens to reflect the performance in real applications.

In addition to the features of the morpheme-context model, we incorporate two priors which capture additional intuitions about morphological segmentations. First, we observe that the number of distinct morphemes used to segment a corpus should be small. This is achieved when the same morphemes are re-used across many different words. Our model incorporates this intuition by imposing a **lexicon prior**: an exponential prior with negative weight on the length of the morpheme lexicon. We define the lexicon to be the set of unique morphemes identified by a complete segmentation of the corpus, and the lexicon length to be the total number of characters in the lexicon. In this way, we can simultaneously emphasize that a lexicon should contain few unique morphemes, and that those morphemes should be short. However, the lexicon prior alone incorrectly favors the trivial segmentation that shatters each word into characters, which results in the smallest lexicon possible (single characters). Therefore, we also impose a **corpus prior**: an exponential prior on the number of mor-

[2] The word as a whole is also treated as a morpheme in itself.

phemes used to segment each word in the corpus, which penalizes over-segmentation. We notice that longer words tend to have more morphemes. Therefore, each word's contribution to this prior is normalized by the word's length in characters (e.g., the segmented word *w-vlAv-wn* contributes 3/7 to the total corpus size). Notice that it is straightforward to incorporate such a prior in a log-linear model, but much more challenging to do so in a directed generative model. These two priors are inspired by the minimum description length (MDL) length principle; the lexicon prior favors fewer morpheme types, whereas the corpus prior favors fewer morpheme tokens. They are vital to the success of our model, providing it with the initial inductive bias.

We also notice that often a word is decomposed into a stem and some prefixes and suffixes. This is particularly true for languages with predominantly inflectional morphology, such as Arabic, Hebrew, and English. Thus our model uses separate lexicons for prefixes, stems, and suffixes. This results in a small but non-negligible accuracy gain in our experiments. We require that a stem contain at least two characters and no fewer characters than any affixes in the same word.[3] In a given word, when a morpheme is identified as the stem, any preceding morpheme is identified as a prefix, whereas any following morpheme as a suffix. The sample segmented corpus mentioned earlier induces the following lexicons:

Prefix w Al
Stem hnAk vlAv bn ywm jmAEp
Suffix wn w

Before presenting our formal model, we first introduce some notation. Let W be a corpus (i.e., a set of words), and S be a segmentation that breaks each word in W into prefixes, a stem, and suffixes. Let σ be a string (character sequence). Each occurrence of σ will be in the form of $\psi_1\sigma\psi_2$, where ψ_1, ψ_2 are the adjacent character n-grams, and $c = (\psi_1, \psi_2)$ is the context of σ in this occurrence. Thus a segmentation can be viewed as a set of morpheme strings and their contexts. For a string x, $L(x)$ denotes the number of characters in x; for a word w, $M_S(w)$ denotes the number of morphemes in w given the segmentation S; $Pref(W,S), Stem(W,S), Suff(W,S)$ denote the lexicons of prefixes, stems, and suffixes induced by S for W. Then, our model defines a joint probability distribution over a restricted set of W and S:

$$P_\theta(W,S) \;=\; \frac{1}{Z}\cdot u_\theta(W,S)$$

where

$$\begin{aligned} u_\theta(W,S) \;=\; \exp\Big(& \sum_\sigma \lambda_\sigma f_\sigma(S) + \sum_c \lambda_c f_c(S) \\ & + \alpha \cdot \sum_{\sigma\in Pref(W,S)} L(\sigma) \\ & + \alpha \cdot \sum_{\sigma\in Stem(W,S)} L(\sigma) \\ & + \alpha \cdot \sum_{\sigma\in Suff(W,S)} L(\sigma) \\ & + \beta \cdot \sum_{w\in W} M_S(w)/L(w) \;\Big) \end{aligned}$$

Here, $f_\sigma(S)$ and $f_c(S)$ are respectively the occurrence counts of morphemes and contexts under S, and $\theta = (\lambda_\sigma, \lambda_c : \sigma, c)$ are their feature weights. α, β are the weights for the priors. Z is the normalization constant, which sums over a set of corpora and segmentations. In the next section, we will define this set for our model and show how to efficiently perform learning and inference.

4 Unsupervised Learning

As mentioned in Smith & Eisner (2005), learning with probabilistic models can be viewed as moving probability mass to the observed data. The question is from where to take this mass. For log-linear models, the answer amounts to defining the set that Z sums over. We use contrastive estimation and define the set to be a neighborhood of the observed data. The instances in the neighborhood can be viewed as pseudo-negative examples, and learning seeks to discriminate them from the observed instances.

Formally, let W^* be the observed corpus, and let $N(\cdot)$ be a function that maps a string to a set of strings; let $N(W^*)$ denote the set of all corpora that can be derived from W^* by replacing every word $w \in W^*$ with one in $N(w)$. Then,

$$Z = \sum_{W\in N(W^*)} \sum_S u(W,S).$$

[3] In a segmentation where several morphemes have the maximum length, any of them can be identified as the stem, each resulting in a distinct segmentation.

Unsupervised learning maximizes the log-likelihood of observing W^*

$$L_\theta(W^*) = \log \sum_S P(W^*, S)$$

We use gradient descent for this optimization; the partial derivatives for feature weights are

$$\frac{\partial}{\partial \lambda_i} L_\theta(W^*) = E_{S|W^*}[f_i] - E_{S,W}[f_i]$$

where i is either a string σ or a context c. The first expected count ranges over all possible segmentations while the words are fixed to those observed in W^*. For the second expected count, the words also range over the neighborhood.

Smith & Eisner (2005) considered various neighborhoods for unsupervised POS tagging, and showed that the best neighborhoods are `TRANS1` (transposing any pair of adjacent words) and `DELORTRANS1` (deleting any word or transposing any pair of adjacent words). We can obtain their counterparts for morphological segmentation by simply replacing "words" with "characters". As mentioned earlier, the instances in the neighborhood serve as pseudo-negative examples from which probability mass can be taken away. In this regard, `DELORTRANS1` is suitable for POS tagging since deleting a word often results in an ungrammatical sentence. However, in morphology, a word less a character is often a legitimate word too. For example, deleting *l* from the Hebrew word *lyhwh* (to the lord) results in *yhwh* (the lord). Thus `DELORTRANS1` forces legal words to compete against each other for probability mass, which seems like a misguided objective. Therefore, in our model we use `TRANS1`. It is suited for our task because transposing a pair of adjacent characters usually results in a non-word.

To combat overfitting in learning, we impose a Gaussian prior (L_2 regularization) on all weights.

5 Supervised Learning

Our learning algorithm can be readily applied to supervised or semi-supervised learning. Suppose that gold segmentation is available for some words, denoted as S^*. If S^* contains gold segmentations for all words in W, we are doing supervised learning; otherwise, learning is semi-supervised. Training now maximizes $L_\theta(W^*, S^*)$; the partial derivatives become

$$\frac{\partial}{\partial \lambda_i} L_\theta(W^*, S^*) = E_{S|W^*,S^*}[f_i] - E_{S,W}[f_i]$$

The only difference in comparison with unsupervised learning is that we fix the known segmentation when computing the first expected counts. In Section 7.3, we show that when labels are available, our model also learns much more effectively than a directed graphical model.

6 Inference

In Smith & Eisner (2005), the objects (sentences) are independent from each other, and exact inference is tractable. In our model, however, the lexicon prior renders all objects (words) interdependent in terms of segmentation decisions. Consider the simple corpus with just two words: *Alrb, lAlrb*. If *lAlrb* is segmented into *l-Al-rb*, *Alrb* can be segmented into *Al-rb* without paying the penalty imposed by the lexicon prior. If, however, *lAlrb* remains a single morpheme, and we still segment *Alrb* into *Al-rb*, then we introduce two new morphemes into the lexicons, and we will be penalized by the lexicon prior accordingly. As a result, we must segment the whole corpus jointly, making exact inference intractable. Therefore, we resort to approximate inference. To compute $E_{S|W^*}[f_i]$, we use Gibbs sampling. To derive a sample, the procedure goes through each word and samples the next segmentation conditioned on the segmentation of all other words. With m samples $S_1, \cdots, S_m$, the expected count can be approximated as

$$E_{S|W^*}[f_i] \approx \frac{1}{m} \sum_j f_i(S_j)$$

There are 2^{n-1} ways to segment a word of n characters. To sample a new segmentation for a particular word, we need to compute conditional probability for each of these segmentations. We currently do this by explicit enumeration.[4] When n is large,

[4] These segmentations could be enumerated implicitly using the dynamic programming framework employed by semi-Markov CRFs (Sarawagi and Cohen, 2004). However, in such a setting, our lexicon prior would likely need to be approximated. We intend to investigate this in future work.

this is very expensive. However, we observe that the maximum number of morphemes that a word contains is usually a small constant for many languages; in the Arabic Penn Treebank, the longest word contains 14 characters, but the maximum number of morphemes in a word is only 5. Therefore, we impose the constraint that a word can be segmented into no more than k morphemes, where k is a language-specific constant. We can determine k from prior knowledge or use a development set. This constraint substantially reduces the number of segmentation candidates to consider; with $k = 5$, it reduces the number of segmentations to consider by almost 90% for a word of 14 characters.

$E_{S,W}[f_i]$ can be computed by Gibbs sampling in the same way, except that in each step we also sample the next word from the neighborhood, in addition to the next segmentation.

To compute the most probable segmentation, we use deterministic annealing. It works just like a sampling algorithm except that the weights are divided by a *temperature*, which starts with a large value and gradually drops to a value close to zero. To make burn-in faster, when computing the expected counts, we initialize the sampler with the most probable segmentation output by annealing.

7 Experiments

We evaluated our system on two datasets. Our main evaluation is on a multi-lingual dataset constructed by Snyder & Barzilay (2008a; 2008b). It consists of 6192 short parallel phrases in Hebrew, Arabic, Aramaic (a dialect of Arabic), and English. The parallel phrases were extracted from the Hebrew Bible and its translations via word alignment and post-processing. For Arabic, the gold segmentation was obtained using a highly accurate Arabic morphological analyzer (Habash and Rambow, 2005); for Hebrew, from a Bible edition distributed by Westminster Hebrew Institute (Groves and Lowery, 2006). There is no gold segmentation for English and Aramaic. Like Snyder & Barzilay, we evaluate on the Arabic and Hebrew portions only; unlike their approach, our system does not use any bilingual information. We refer to this dataset as **S&B** . We also report our results on the Arabic Penn Treebank (**ATB**), which provides gold segmentations for an Arabic corpus with about 120,000 Arabic words.

As in previous work, we report recall, precision, and F1 over segmentation points. We used 500 phrases from the S&B dataset for feature development, and also tuned our model hyperparameters there. The weights for the lexicon and corpus priors were set to $\alpha = -1$, $\beta = -20$. The feature weights were initialized to zero and were penalized by a Gaussian prior with $\sigma^2 = 100$. The learning rate was set to 0.02 for all experiments, except the full Arabic Penn Treebank, for which it was set to 0.005.[5] We used 30 iterations for learning. In each iteration, 200 samples were collected to compute each of the two expected counts. The sampler was initialized by running annealing for 2000 samples, with the temperature dropping from 10 to 0.1 at 0.1 decrements. The most probable segmentation was obtained by running annealing for 10000 samples, using the same temperature schedule. We restricted the segmentation candidates to those with no greater than five segments in all experiments.

7.1 Unsupervised Segmentation on S&B

We followed the experimental set-up of Snyder & Barzilay (2008b) to enable a direct comparison. The dataset is split into a training set with 4/5 of the phrases, and a test set with the remaining 1/5. First, we carried out unsupervised learning on the training data, and computed the most probable segmentation for it. Then we fixed the learned weights and the segmentation for training, and computed the most probable segmentation for the test set, on which we evaluated.[6] Snyder & Barzilay (2008b) compared several versions of their systems, differing in how much bilingual information was used. Using monolingual information only, their system (S&B-MONO) trails the state-of-the-art system Morfessor; however, their best system (S&B-BEST), which uses bilingual information that includes phrasal alignment and phonetic correspondence between Arabic and Hebrew, outperforms Morfessor and achieves the state-of-the-art results on this dataset.

[5] The ATB set is more than an order of magnitude larger and requires a smaller rate.

[6] With unsupervised learning, we can use the entire dataset for training since no labels are provided. However, this set-up is necessary for S&B's system because they used bilingual information in training, which is not available at test time.

ARABIC	Prec.	Rec.	F1
S&B-MONO	53.0	78.5	63.2
S&B-BEST	67.8	77.3	72.2
FULL	76.0	80.2	**78.1**
HEBREW	Prec.	Rec.	F1
S&B-MONO	55.8	64.4	59.8
S&B-BEST	64.9	62.9	63.9
FULL	67.6	66.1	**66.9**

Table 1: Comparison of segmentation results on the S&B dataset.

Table 1 compares our system with theirs. Our system outperforms both S&B-MONO and S&B-BEST by a large margin. For example, on Arabic, our system reduces F1 error by 21% compared to S&B-BEST, and by 40% compared to S&B-MONO. This suggests that the use of monolingual morpheme context, enabled by our log-linear model, is more helpful than their bilingual cues.

7.2 Ablation Tests

To evaluate the contributions of the major components in our model, we conducted seven ablation tests on the S&B dataset, each using a model that differed from our full model in one aspect. The first three tests evaluate the effect of priors, whereas the next three test the effect of context features. The last evaluates the impact of using separate lexicons for affixes and stems.

NO-PRIOR The priors are not used.
NO-COR-PR The corpus prior is not used.
NO-LEX-PR The lexicon prior is not used.
NO-CONTEXT Context features are not used.
UNIGRAM Unigrams are used in context.
BIGRAM Bigrams are used in context.
SG-LEXICON A single lexicon is used, rather than three distinct ones for the affixes and stems.

Table 2 presents the ablation results in comparison with the results of the full model. When some or all priors are excluded, the F1 score drops substantially (over 10 points in all cases, and over 40 points in some). In particular, excluding the corpus prior, as in NO-PRIOR and NO-COR-PR, results in over-segmentation, as is evident from the high recalls and low precisions. When the corpus prior is enacted but not the lexicon priors (NO-LEX-PR), precision is much higher, but recall is low; the system now errs on under-segmentation because recurring strings are often not identified as morphemes.

ARABIC	Prec.	Rec.	F1
FULL	76.0	80.2	**78.1**
NO-PRIOR	24.6	89.3	38.6
NO-COR-PR	23.7	87.4	37.2
NO-LEX-PR	79.1	51.3	62.3
NO-CONTEXT	71.2	62.1	66.3
UNIGRAM	71.3	76.5	73.8
BIGRAM	73.1	78.4	75.7
SG-LEXICON	72.8	82.0	77.1
HEBREW	Prec.	Rec.	F1
FULL	67.6	66.1	66.9
NO-PRIOR	34.0	89.9	49.4
NO-COR-PR	35.6	90.6	51.1
NO-LEX-PR	65.9	49.2	56.4
NO-CONTEXT	63.0	47.6	54.3
UNIGRAM	63.0	63.7	63.3
BIGRAM	69.5	66.1	**67.8**
SG-LEXICON	67.4	65.7	66.6

Table 2: Ablation test results on the S&B dataset.

A large accuracy drop (over 10 points in F1 score) also occurs when the context features are excluded (NO-CONTEXT), which underscores the importance of these overlapping features. We also notice that the NO-CONTEXT model is comparable to the S&B-MONO model; they use the same feature types, but different priors. The accuracies of the two systems are comparable, which suggests that we did not sacrifice accuracy by trading the more complex and restrictive Dirichlet process prior for exponential priors. A priori, it is unclear whether using contexts larger than unigrams would help. While potentially beneficial, they also risk aggravating the data sparsity and making our model more prone to overfitting. For this problem, however, enlarging the context (using higher n-grams up to trigrams) helps substantially. For Arabic, the highest accuracy is attained by using trigrams, which reduces F1 error by 16% compared to unigrams; for Hebrew, by using bigrams, which reduces F1 error by 17%. Finally, it helps to use separate lexicons for affixes and stems, although the difference is small.

ARABIC	%Lbl.	Prec.	Rec.	F1
S&B-MONO-S	100	73.2	92.4	81.7
S&B-BEST-S	200	77.8	92.3	84.4
FULL-S	25	84.9	85.5	85.2
	50	88.2	86.8	87.5
	75	89.6	86.4	87.9
	100	91.7	88.5	**90.0**
HEBREW	%Lbl.	Prec.	Rec.	F1
S&B-MONO-S	100	71.4	79.1	75.1
S&B-BEST-S	200	76.8	79.2	78.0
FULL-S	25	78.7	73.3	75.9
	50	82.8	74.6	78.4
	75	83.1	77.3	80.1
	100	83.0	78.9	**80.9**

Table 3: Comparison of segmentation results with supervised and semi-supervised learning on the S&B dataset.

7.3 Supervised and Semi-Supervised Learning

To evaluate our system in the supervised and semi-supervised learning settings, we report the performance when various amounts of labeled data are made available during learning, and compare them to the results of Snyder & Barzilay (2008a). They reported results for supervised learning using monolingual features only (S&B-MONO-S), and for supervised bilingual learning with labels for both languages (S&B-BEST-S). On both languages, our system substantially outperforms both S&B-MONO-S and S&B-BEST-S. E.g., on Arabic, our system reduces F1 errors by 46% compared to S&B-MONO-S, and by 36% compared to S&B-BEST-S. Moreover, with only one-fourth of the labeled data, our system already outperforms S&B-MONO-S. This demonstrates that our log-linear model is better suited to take advantage of supervised labels.

7.4 Arabic Penn Treebank

We also evaluated our system on the Arabic Penn Treebank (ATB). As is common in unsupervised learning, we trained and evaluated on the entire set. We compare our system with Morfessor (Creutz and Lagus, 2007).[7] In addition, we compare with Morfessor Categories-MAP, which builds on Morfessor and conducts an additional greedy search specifically tailored to segmentation. We found that it performs much better than Morfessor on Arabic but worse on Hebrew. To test each system in a low-data setting, we also ran experiments on the set containing the first 7,000 words in ATB with at least two characters (ATB-7000). Table 4 shows the results. Morfessor performs rather poorly on ATB-7000. Morfessor Categories-MAP does much better, but its performance is dwarfed by our system, which further cuts F1 error by half. On the full ATB dataset, Morfessor performs even worse, whereas Morfessor Categories-MAP benefits from the larger dataset and achieves an F1 of 74.9. Still, our system substantially outperforms it, further reducing F1 error by 11%.[8]

ATB-7000	Prec.	Rec.	F1
MORFESSOR-1.0	70.6	34.3	46.1
MORFESSOR-MAP	86.9	46.4	60.5
FULL	83.4	77.3	**80.2**
ATB	Prec.	Rec.	F1
MORFESSOR-1.0	80.7	20.4	32.6
MORFESSOR-MAP	77.4	72.6	74.9
FULL	88.5	69.2	**77.7**

Table 4: Comparison of segmentation results on the Arabic Penn Treebank.

8 Conclusion

This paper introduces the first log-linear model for unsupervised morphological segmentation. It leverages overlapping features such as morphemes and their contexts, and enables easy extension to incorporate additional features and linguistic knowledge. For Arabic and Hebrew, it outperforms the state-of-the-art systems by a large margin. It can also be readily applied to supervised or semi-supervised learning when labeled data is available. Future directions include applying our model to other inflectional and agglutinative languages, modeling internal variations of morphemes, leveraging parallel data in multiple languages, and combining morphological segmentation with other NLP tasks, such as machine translation.

[7] We cannot compare with Snyder & Barzilay's system as its strongest results require bilingual data, which is not available.

[8] Note that the ATB and ATB-7000 experiments each measure accuracy on their entire training set. This difference in testing conditions explains why some full ATB results are lower than ATB-7000.

References

Galen Andrew. 2006. A hybrid markov/semi-markov conditional random field for sequence segmentation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Michael R. Brent, Sreerama K. Murthy, and Andrew Lundberg. 1995. Discovering morphemic suffixes: A case study in minimum description length induction. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society*.

Tim Buckwalter. 2004. Buckwalter Arabic morphological analyzer version 2.0.

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

Sajib Dasgupta and Vincent Ng. 2007. High-performance, language-independent morphological segmentation. In *Proceedings of Human Language Technology (NAACL)*.

Vera Demberg. 2007. A language-independent unsupervised model for morphological segmentation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems 18*.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2007. Distributional cues to word segmentation: Context is important. In *Proceedings of the 31st Boston University Conference on Language Development*.

Alan Groves and Kirk Lowery, editors. 2006. *The Westminster Hebrew Bible Morphology Database*. Westminster Hebrew Institute, Philadelphia, PA, USA.

Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.

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

Samarth Keshava and Emily Pitler. 2006. A simple, intuitive approach to morpheme induction. In *Proceedings of 2nd Pascal Challenges Workshop*, Venice, Italy.

Dan Klein and Christopher D. Manning. 2001. Natural language grammar induction using a constituent-context model. In *Advances in Neural Information Processing Systems 14*.

Mikko Kurimo, Mathias Creutz, and Ville Turunen. 2007. Overview of Morpho Challenge in CLEF 2007. In *Working Notes of the CLEF 2007 Workshop*.

Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with markov logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 649–658, Honolulu, HI. ACL.

Ronald Rosenfeld. 1997. A whole sentence maximum entropy language model. In *IEEE workshop on Automatic Speech Recognition and Understanding*.

Sunita Sarawagi and William Cohen. 2004. Semimarkov conditional random fields for information extraction. In *Proceedings of the Twenty First International Conference on Machine Learning*.

Patrick Schone and Daniel Jurafsky. 2001. Knowlege-free induction of inflectional morphologies. In *Proceedings of Human Language Technology (NAACL)*.

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.

Benjamin Snyder and Regina Barzilay. 2008a. Cross-lingual propagation for morphological analysis. In *Proceedings of the Twenty Third National Conference on Artificial Intelligence*.

Benjamin Snyder and Regina Barzilay. 2008b. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*.