

---

# Unsupervised Morphological Disambiguation using Statistical Language Models

---

**Mehmet Ali Yatbaz**  
Dept. of Computer Engineering  
Koç University  
İstanbul, Turkey  
myatbaz@ku.edu.tr

**Deniz Yuret**  
Dept. of Computer Engineering  
Koç University  
İstanbul, Turkey  
dyuret@ku.edu.tr

## Abstract

In this paper, we present a probabilistic model for the unsupervised morphological disambiguation problem. Our model assigns morphological parses  $T$  to the contexts  $C$  instead of assigning them to the words  $W$ . The target word  $w \in W$  determines the possible parse set  $T_w \subset T$  that can be used in  $w$ 's context  $c_w \in C$ . To assign the correct morphological parse  $t \in T_w$  to  $w$ , our model finds the parse  $t \in T_w$  that maximizes  $P(t|c_w)$ .  $P(t|c_w)$ 's are estimated using a statistical language model and the vocabulary of the corpus. The system performs significantly better than an unsupervised baseline and its performance is close to a supervised baseline.

## 1 Introduction

The morphological disambiguation problem can be defined as selecting the correct parse of a word in a given context from the possible candidate parses of the word. Our approach does not directly assign the parses to the target word, instead it uses the target word to limit the set of possible parses and then assigns probabilities for these using the context. This approach has been previously applied to the word sense disambiguation problem where the aim is to determine the sense of an ambiguous word in a given context [6].

The main challenge of the supervised morphological disambiguation is the difficulty of acquiring a sufficient amount of morphologically parsed training data. For example, the largest available Turkish morphology training data is a corpus of 1 million semi-automatically tagged words and due to the semi-automatic tagging the training data itself has inconsistencies. In contrast, our model estimates the necessary probabilities by using an unsupervised model, therefore it is not affected by the tagged data bottleneck and inconsistencies. The untagged text data we use consists of a 440 million word Turkish corpus which is derived from a variety of domains while the supervised data is a corpus of 1 million words from a specific news domain. Another issue is, unlike English, in agglutinative languages the number of theoretically possible parses can be infinite although the number of features is finite. Therefore, even in a training corpus of 1 million words it is possible to observe thousands of different possible parses which leads to data sparseness. Finally, our model can be applied to any agglutinative language since it does not require any hand-crafted rules or the knowledge of a native speaker.

To predict the correct parse of an ambiguous word, first the possible parses are generated using a morphological analyzer. Then using the language model together with the vocabulary of the corpus, a probabilistic model is applied to each ambiguous word. The resulting disambiguation accuracy for the ambiguous words is 64.5% where 31.9% and 71.0% are the unsupervised and supervised baselines respectively.

Morphological disambiguation is an important step for a number of NLP tasks and this importance becomes more crucial for agglutinative languages such as Turkish, Finnish, Hungarian and Czech. For example, by using a morphological analyzer together with a disambiguator the perplexity of a Turkish language model can be reduced significantly [9].

Below you can see three possible morphological parses for the Turkish word “*masali*”. The candidate parses are generated using a morphological analyzer. The first token of the analyzer output is

masal	+Noun+A3sg+Pnon+Acc	(= the story)
masal	+Noun+A3sg+P3sg+Nom	(= his story)
masa	+Noun+A3sg+Pnon+Nom <sup>^</sup> DG+Adj+With	(= with tables)

the root of the word while the rest is the parse of the word that consists of features that are concatenated to each other either by a “+” or “<sup>^</sup>DG”. The first two lines of the analyzer output for “*masali*” have the same root, *masal* (= story) but different parses while the last one has a different root *masa* (= table) and parse. Feature groups that are separated by a derivation boundary (<sup>^</sup>DG) are called inflection groups [5]. The first feature following the root or a <sup>^</sup>DG represents the part-of-speech (POS) tag of the new derived word. A morphological disambiguation system should pick the correct parse of the word “*masali*” given the context in which it appears.

The next section will describe the model, parameter estimation and the algorithm for our unsupervised morphological disambiguator. Section 3 presents the experiments and results. Section 4 introduces the related work while Section 5 concludes the paper.

## 2 Unsupervised Morphological Disambiguator

### 2.1 Model

Our model is built on the idea of assigning morphological parses to the word contexts instead of the word itself. Therefore, it selects the parse  $t$  of the target word  $w$  that is most likely in the target word context,  $c_w$ . The model finds the parse  $t$  in the set of possible parses of the target word  $T_w$  that maximizes  $P(t|c_w)$  which is the probability of a parse in a given context  $c_w$ . This probability is calculated by using possible replacement words from the vocabulary  $V$ . Our model can be written as,

$$\operatorname{argmax}_{t \in T_w} P(t|c_w) = \sum_{v \in V} P(t|v, c_w) P(v|c_w) \quad (1)$$

### 2.2 Estimation

In Section 2.1, we showed that our model is decomposed into the estimation of  $P(v|c_w)$  and  $P(t|v, c_w)$ . We estimate  $P(v|c_w)$  using a statistical language model. We use two assumptions when estimating  $P(t|v, c_w)$ .

1. **Pruning Assumption:** Every  $w$  has a possible parse set  $T_w$  which is produced by the morphological analyzer. Instead of assigning non-zero probabilities to all possible parses, our model simply assumes that in the context of  $w$  only possible parses are the ones that are contained in  $T_w$ . Therefore, parses that are not in  $T_w$  have zero probability.
2. **Uniformity Assumption:** We assume the distribution of the parses given a substitute word  $v$  and  $c_w$  is uniform on  $T_w$ .

$$P(t|v, c_w) = \begin{cases} \frac{1}{|T_w \cap T_v|} & \text{if } t \in T_w \cap T_v, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

In other words, we assume that  $P(v|c_w)$  is shared equally among the common parses of target word  $w$  and the replacement word  $v$ .

To estimate  $P(v|c_w)$ , the distribution of the target word replacements in a given context, we use an n-gram language model. The context is defined as the  $2n - 1$  word window  $w_{-n+1} \dots w_0 \dots w_{n-1}$  and it is centered at the target word position. The probability of a word in a given context can be estimated as:

$$P(w_0 = v) \propto P(w_{-n+1} \dots w_0 \dots w_{n-1}) \quad (3)$$

$$= P(w_{-n+1})P(w_{-n+2}|w_{-n+1}) \dots P(w_{n-1}|w_{-n+1}^{n-2}) \quad (4)$$

$$\propto P(w_0|w_{-n+1}^{-1}) \dots P(w_1|w_{-n+2}^0) \dots P(w_{n-1}|w_0^{n-2}) \quad (5)$$

where  $w_i^j$  represents the sequence of words  $w_i w_{i+1} \dots w_j$ . In Equation 3,  $P(v|c_w)$  is proportional to  $P(w_{-n+1} \dots w_0 \dots w_{n-1})$  since the context of the target word replacements is fixed. In Equation 4, terms without  $v$  are common for every replacement therefore they have been dropped. Finally, because of the Markov property of n-gram language model, only  $n - 1$  words are used as a conditional context.

The probabilities in Equation 5 are calculated using a language model that is trained on the Turkish corpus described in [7]. The data set contains about 440 million words and 10% of the data is split and used as the test set to calculate the perplexity of the language models. The SRILM toolkit is used to train n-gram models with different smoothing methods, n-gram orders and training corpus sizes. The affect of each model on the performance of the algorithm is detailed in Section 3.

### 2.2.1 Parse Simplification

	Original Parse	Simplified Parse
masal	+Noun+A3sg+Pnon+Acc	Pnon+Acc
masal	+Noun+A3sg+P3sg+Nom	P3sg+Nom
masa	+Noun+A3sg+Pnon+Nom ^ DG+Adj+With	With

Table 1: Parse simplification of the word “*masali*”.

The estimation quality of  $P(t|c_w)$  highly depends on the parse set  $T_w$  of the target word. If the number of replacement words that have common parses with the target word is small then  $P(t|c_w)$  will be estimated using very few replacement words. Thus, instead of using the parses directly, we construct a discriminative minimal feature set  $S_w$  of  $T_w$  from the final inflection groups of each parse. To construct  $S_w$ , our model selects the minimum number of rightmost features from each of the last IG’s such that these rightmost features uniquely discriminate the corresponding parse from the other parses in  $T_w$ . Table 1 represents an example simplification of the parses of the word “*masali*”.

### 2.3 Algorithm

Section 2.1 described the mathematical framework of the model applied to the morphological disambiguation problem. In this section, the algorithmic steps of the disambiguator is presented and throughout this section  $w_i$  denotes the  $i^{th}$  word from the set of target words  $W$ ,  $c_i$  denotes the context of  $i^{th}$  target word,  $v_{ij}$  denotes the  $j^{th}$  replacement of  $i^{th}$  target word,  $T_{w_i}$  denotes the set of possible morphological parses of  $w_i$  and  $V_k$  denotes the set of  $k$  most frequent words of training corpus vocabulary.

Steps of the Algorithm:

1. Construct a morphological dictionary for all the words in the vocabulary by using the morphological analyzer and construct  $V_k$ .
2. Construct  $S_{w_i}$  by simplifying the  $T_{w_i}$  according to the Section 2.2.1.
3. Calculate  $P(v_{ij}|c_i)$  of each replacement using the estimation method described in Section 2.2.
4. Calculate  $P(t|c_i)$  for all  $t \in S_{w_i}$  using  $P(v_{ij}|c_i)$  that are calculated in Step 3.  $P(t|v_{ij})$  is equal to  $\frac{1}{|S_{w_i} \cap S_{v_{ij}}|}$  due to the uniformity assumption.
5. Select  $t \in S_i$  that maximizes  $P(t|c_i)$ .

### 3 Experiments and Results

	Test Set	Tagged Trained Set
Sentences	446	50673
Tokens	5365	948404
Ambiguous tokens	2437(45.4%)	399223(42.1%)
Average Parses	1.85	1.76

Table 2: Test and Tagged Train Data Statistics

In this section we present a number of experiments to observe the effects of the model parameters on the algorithm performance. We define an unsupervised and a supervised baseline on the test set to compare with the results of our method. The unsupervised baseline is calculated by randomly picking one of the parses of each word in the test set. To calculate a supervised baseline, we use a tagged training set that consists of 1 million words of semi-automatically disambiguated Turkish news text. Some brief statistics relevant to tagged training set and the test set are presented in Table 2. The supervised baseline simply does majority voting for each word using the training set. If the target word does not exist in the training set, the supervised baseline randomly picks one of the possible parses of the missing word. The unsupervised baseline disambiguates 39.4% of the ambiguous words correctly while the supervised baseline correctly disambiguates 71.0% of them. All the accuracy scores that are reported in this section include only the ambiguous words. The experiments in this section can be categorized as the corpus size experiments and the replacement size experiments<sup>0</sup>.

#### 3.1 Corpus size

We used three corpora with different sizes to train the 4-gram language model and observe the performance of our disambiguator. In order to do the experiments, we randomly select 1% and 10% of the original training corpus detailed in Section 2.2. The performance of the disambiguator with different sized corpora, are summarized in Table 3.

Corpus Size	Accuracy
4M	60.4
40M	63.1
400M	64.5

Table 3: The performance of the model using the second replacement routine together with different parameter settings. The 95% confidence interval for each result is  $\pm 1.9$ .

As Table 3 shows, the performance decreases as the corpus size becomes smaller. However, using 10% of the corpus our disambiguator can still achieve comparable results (in terms of 95% confidence interval) with the model using the whole corpus. This is not the case when we use 1% of the corpus, since the loss of performance compared to the model using whole corpus is statistically significant. These experiments show that, the performance may be improved by using a larger Turkish corpora.

We used the Good-Turing and the Kneser-Ney smoothing techniques to observe the effect of smoothing on the probability estimates of our disambiguator, however we found that the choice of smoothing method does not significantly affect the model performance. Similarly, 2, 3 and 4-gram language models were trained, however they did not have any significant effect on the performance of our model.

#### 3.2 Number of Replacement Words

In these experiments, we calculate  $P(v|c_w)$  of each replacement word and select 10, 100, 200 and 2000 replacement words that have the highest  $P(v|c_w)$  and use only these words to estimate

<sup>0</sup>Unless otherwise stated, for the sake of simplicity all the reported results in this section are obtained by using the most frequent 200K words of the vocabulary.

Number of replacements	Accuracy
Top 10	63.4
Top 100	64.3
Top 200	64.4
Top 2000	64.5

Table 4: The performance of the model with different number of replacements using the second replacement routine. The 95% confidence interval for each accuracy in this table is  $\pm 1.9$  by scientific rounding.

$P(t|c_w)$ . Table 4 shows the performance of each case with different settings. By using the 95% confidence interval, the results of each model with different number of replacements are not significantly different. Thus, computational efficiency of our model can be increased by using a possibly faster algorithm that heuristically finds the top  $k$  replacement words with the highest  $P(v|c_w)$ .

## 4 Related Work

Several studies have made progress on the unsupervised morphological disambiguation of the morphologically rich languages in the past decade.

In Hebrew, a context free model was used to estimate the morpho-lexical probabilities of a given word from an untagged corpus [3]. Similar to Turkish, Hebrew is a morphologically rich language and morphemes in Hebrew can combine into a single word in both agglutinative and fusional ways. Thus a Hebrew word can have various segmentations and morphological analyses. This method is very similar to ours because both use replacement words to disambiguate the target word. Our method uses one set of replacement words from the vocabulary while [3] explicitly uses a predefined set of rules to select the set of similar words for each target word before the disambiguation task. Another important difference is, [3] do not use any contextual information during the disambiguation task.

A more recent study has shown that morpheme-based segmentation and tagging in Hebrew can be learned simultaneously by using a stochastic unsupervised learning with HMM [1]. Their model first estimates the probabilities of each segmentation and their possible tags by using a variation of the Baum-Welch algorithm. Then an adaptation of the Viterbi algorithm is applied to get the most probable segmentation and tagging sequence.

The morphological disambiguation task in English has been covered under the part-of-speech task due to the simpler morphological structure of the language. Previous well known studies on the unsupervised POS disambiguation of English include a hidden Markov model (HMM) that was trained on unlabeled English text by using the maximum likelihood estimator (MLE) with different initializations[4]. A more recent work has shown that instead of using HMM together with the expectation maximization (EM), one can use conditional random fields that are estimated using contrastive estimation which outperformed the models trained with EM [8]. In the non-parametric Bayesian approach, Goldwater and Griffiths used a fully Bayesian HMM model that averages over possible parameter values. Their model outperformed the model with ML estimation and achieved comparable results with the state-of-the-art discriminative models [2].

## 5 Conclusion and Future Work

In this paper, we have presented an unsupervised probabilistic model for the morphological disambiguation task of Turkish. The main idea behind our model is instead of assigning parses to words, it assigns parses to the contexts of the words. The probability of the morphological analysis in a given context is estimated by a language model that is trained on an unlabeled corpus. Therefore, the model does not require any predefined rule set and it can be applied to any language as long as a parse dictionary for each word and a corpus are available. We were able to achieve 64.5% accuracy using this model. This accuracy might be improved by relaxing the uniformity assumption of the target word parse distribution and letting it to converge to the actual probabilities by using better statistical inference methods.

## References

- [1] M. Adler and M. Elhadad. An unsupervised morpheme-based hmm for hebrew morphological disambiguation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 665–672, 2006.
- [2] S. Goldwater and T. Griffiths. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Annual Meeting-Assosiation for Computational Linguistics*, volume 45, page 744, 2007.
- [3] M. Levinger, A. Itai, and U. Ornan. Learning morpho-lexical probabilities from an untagged corpus with an application to Hebrew. *Computational Linguistics*, 21(3):404, 1995.
- [4] B. Merialdo. Tagging english text with a probabilistic model. *Computational linguistics*, 20(2):155–171, 1994.
- [5] K. OflazerH, D.Z. Hakkani-Tür, and G. Tür. Design for a Turkish treebank.
- [6] J. Pustejovsky, P. Hanks, and A. Rumshisky. Automated induction of sense in context. In *Proceedings of the 20th international conference on Computational Linguistics*, page 924. Association for Computational Linguistics, 2004.
- [7] H. Sak, T. Güngör, and M. Saraçlar. Turkish language resources: Morphological parser, morphological disambiguator and web corpus. *Lecture Notes in Computer Science*, 5221:417–427, 2008.
- [8] Noah A. Smith and Jason Eisner. Contrastive estimation: training log-linear models on unlabeled data. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 354–362, Morristown, NJ, USA, 2005. Association for Computational Linguistics.
- [9] D. Yuret and E. Biçici. Modeling Morphologically Rich Languages Using Split Words and Unstructured Dependencies.