A Novel Approach to Morphological Disambiguation for Turkish

Onur Görgün and Olcay Taner Yildiz

Abstract In this paper, we propose a classification based approach to the morphological disambiguation for Turkish language. Due to complex morphology in Turkish, any word can get unlimited number of affixes resulting very large tag sets. The problem is defined as choosing one of parses of a word not taking the existing root word into consideration. We trained our model with well-known classifiers using WEKA toolkit and tested on a common test set. The best performance achieved is 95.61% by J48 Tree classifier.

1 Introduction

Morphological disambiguation problem is defined as the task of selecting the correct morphological parse of a word among its parses. According to the morphophonemic structure of the language and morphotactics which define the ordering of morphemes, a word may have many parses. These parses may share the same root word or may have different root words. Morphological disambiguation is considered as a preliminary step for higher level language analysis.

Turkish is one of the morphologically rich languages. Like other agglutinative languages, due to its free constituent order nature, Turkish has a large number of possible tags. There have been studies for morphological disambiguation problem in Turkish. These studies can be categorized under two main approaches: rule-

O. Görgün (✉) · O. T. Yildiz
Department of Computer Science and Engineering,
Işik University, Şile, 34980 İstanbul, Turkey
e-mail: onurg@isikun.edu.tr

O. T. Yildiz (✉)
e-mail: olcaytaner@isikun.edu.tr

E. Gelenbe et al. (eds.), *Computer and Information Sciences II*,
DOI: 10.1007/978-1-4471-2155-8_9,

based approaches and statistical approaches. In statistical approaches, a large corpus is used to train the statistical model and the trained model is tested on an unseen test corpus [1]. However, due to the large number of tags in Turkish, data sparseness is a serious problem. To cope with the data sparseness problem, morphological parses are divided into smaller parts called inflectional groups [2]. The most recent approach to the morphological disambiguation problem is presented in [3]. The methodology employed is based on ranking of the most possible parse sequences (determined by the baseline statistical model represented in [1]) with Perceptron algorithm. The very early rule-based approach to Turkish used hand-crafted rules [5]. The combination of both rule-based and machine learning approaches also exists such as [4].

In this paper we propose a classification approach to the morphological disambiguation problem. The idea behind our algorithm is as follows: Considering each set of distinct possible parses as a classification problem, one can divide the morphological disambiguation problem into multiple classification problems. Then each classification problem can be solved independently using any machine learning classifier. The inputs (features) of the classification problem are the existence of the part of speech tags in the previous two neighbor words.

The paper is organized as follows: In Sect. 2 we introduce the morphological disambiguation problem and formalize it. In Sect. 3, we will review the previous approaches to the morphological disambiguation problem in Turkish. In Sect. 4 we introduce our proposed approach. We give our experiments results in Sect. 5 and conclude in Sect. 6.

2 Morphological Disambiguation

Morphological disambiguation is the problem of selecting accurate morphological parse of a word given its possible parses. These parses are generated by a morphological analyzer [6, 7]. In morphologically rich languages like Turkish, the number of possible parses for a given word is generally more than one. Each parse is considered as a different interpretation of a single word. Each interpretation consists of a root word and sequence of inflectional and derivational suffixes. Table 1 illustrates different interpretations of the word “üzerine”.

As seen above, the first two parses share the same root but different suffix sequences. Similarly, the last two parses also share the same root, however sequence of morphemes are different. Given a parse such as

$$\text{üz} + \text{Verb} + \text{Pos} + \text{Aor} +^{\wedge} \text{DB} + \text{Adj} + \text{Zero} +^{\wedge} \text{DB} + \text{Noun} + \text{Zero} + \text{A3sg} + \text{P3sg} + \text{Dat}$$

each item is separated by “+” is a morphological feature such as Pos or Aor. Inflectional groups are identified as sequence of morphological features separated by

Table 1 Four possible parses of word "üzerine"

üzer+Noun+A3sg+P3sg+Dat
üzer+Noun+A3sg+P2sg+Dat
üz+Verb+Pos+Aor+^ DB+Adj+Zero+^ DB+Noun+Zero+A3sg+P3sg+Dat
üz+Verb+Pos+Aor+^ DB+Adj+Zero+^ DB+Noun+Zero+A3sg+P2sg+Dat

derivational boundaries ($^\wedge$DB). The sequence of inflectional groups forms the term tag. Root word plus tag is named as word form. So, a word form is defined as follows:

$$\text{IGroot} + \text{IG}_1 + ^\wedge\text{DB} + \text{IG}_2 + ^\wedge\text{DB} + \cdots + ^\wedge\text{DB} + \text{IG}_n$$

Then the morphological disambiguation problem can be defined as follows: For a given sentence represented by a sequence of words $W = w_1^n = w_1, w_2, \ldots, w_n$, determine the sequence of parses $T = t_1^n = t_1, t_2, \ldots, t_n$, where t_i represents the correct parse of the word w_i. Using the Bayesian approach, the problem is formulated as follows:

$$\arg\max_T P(T|W) = \frac{P(T)P(W|T)}{P(W)} \tag{1}$$

where *P(W)* is constant for all $P(W|T)$.$P(W|T)$ is equal to 1, since given a tag sequence, there is only one possible word form corresponding to it. So the morphological disambiguation problem is simplified as the following:

$$\arg\max_T P(T|W) = \arg\max_T P(T) \tag{2}$$

3 Related Work

The baseline model described in [1] generates the most probable tag sequence for a given word sequence using Viterbi decoding. First, they break down the tag from derivation boundaries called inflectional group (IG). The problem is formulated as follows:

$$\begin{aligned} P(T) &= \prod_{i=1}^{n} P(t_i|t_{i-2}, t_{i-1}) \\ &= \prod_{i=1}^{n} ((r_i, IG_{i,1}, \ldots, IG_{i,n_i})| \\ &\quad (r_{i-2}, IG_{i-2,1}, \ldots, IG_{i-2,n_{i-2}}) \\ &\quad (r_{i-1}, IG_{i-1,1}, \ldots, IG_{i-2,n_{i-1}})) \end{aligned} \tag{3}$$

where n_i represents the number of inflectional groups associated with the ith parse and $G_{i,j}$ represents the jth inflectional group of parse i. The baseline trigram-based model is based on two basic assumptions: (1) root of the current word only depends on root of two previous words, and (2) presence of sequence of IGs in the current word depend only the last IG of two previous words. Under these assumptions, $P(T)$ is re-formulated as:

$$P(T) = \prod_{i=1}^{n}((P(r_i|r_{i-2}, r_{i-1}) \prod_{k=1}^{n_i} P(IG_{i,k}|IG_{i-2,n_{i-2}}, IG_{i-1,n_{i-1}}) \quad (4)$$

The trigram probabilities are estimated using standard n-gram probability estimation methods using morphologically disambiguated training data.

The Greedy Prepend Algorithm is a rule-based approach, based on decision lists [4]. Each pattern is formed by surface attributes of surrounding words of the current word. The decision lists are formed for each of the 126 distinct morphological features that exist. In the model, a 5-word (including word W, the first two left and two right neighbors), window is used. Greedy Prepend list reduction algorithm is used to generate the decision lists. The algorithm starts with the most general rule which covers all instances. The algorithm adds rules one by one where the best rule is determined using information gain. The algorithm stops when no improvement can be made.

The Perceptron Algorithm [3] is a combination of statistical and machine learning approaches. They use the Baseline Trigram-Based Model to generate n-best parses for each sentence. A feature set consisting of 23 features is used to disambiguate the current parse. The model also takes into account previous two words. Using the n-best parses as input to the algorithm, the algorithm makes multiple iterations over the training set to estimate parameter values. The highest scoring candidate is then selected using current parameter values. If the highest scoring candidate is different than the correct one, parameter values are updated.

4 Proposed Approach

We define the disambiguation task as identifying the correct parse from N possible parses excluding the root word. Consider the example in Table 2 for the word "üzerine". Our approach defines the classification problem as follows:

Class 1:Noun+A3sg+P3sg+Dat
Class 2:Noun+A3sg+P2sg+Dat
Class 3:Verb+Pos+Aor+^DB+Adj+Zero+^DB+Noun+Zero+A3sg+P3sg+Dat
Class 4:Verb+Pos+Aor+^DB+Adj+Zero+^DB+Noun+Zero+A3sg+P2sg+Dat

where the correct parse is Class 1. Although a word can take theoretically unlimited number of suffixes [5], the number of distinct problems (classification problem) is

Table 2 Distribution of problems with respect to the # of instances

Number of instances	Number of problems
1–10	7213
11–100	1617
101–1000	427
1001–10000	60
10001–100000	3

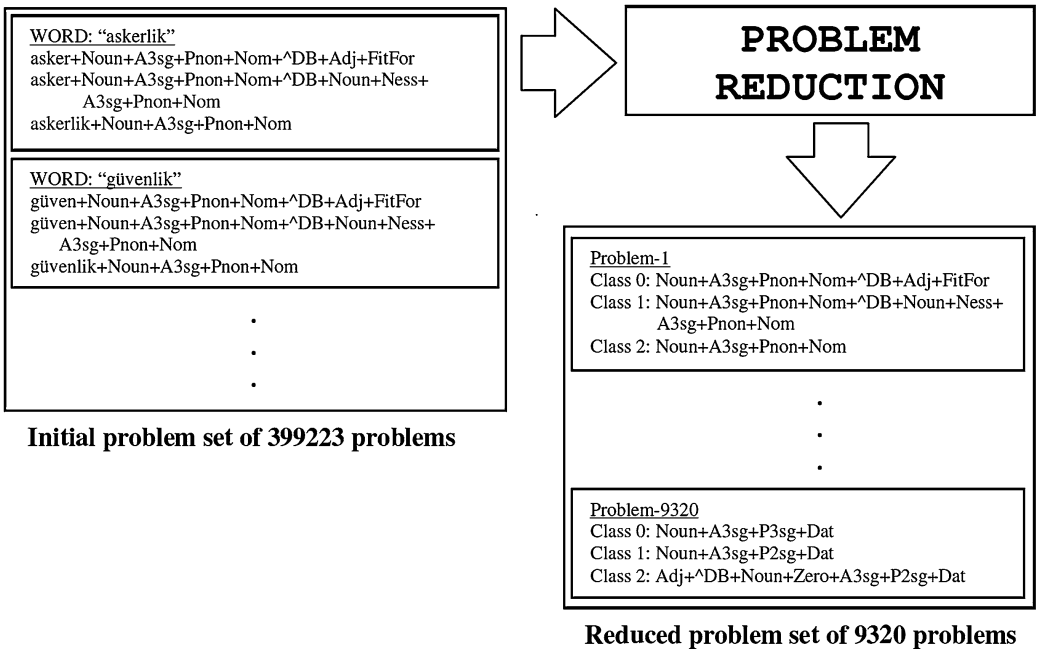

Fig. 1 Problem reduction step of the training phase. The parses of "askerlik" and "güvenlik" map to problem 1

9320 for a 1M size disambiguated train set. If *n* problems are same, only one of them is kept and others are discarded from the problem set. After this preprocessing the distribution of the problems with respect to the number of instances are given in Table 2.

Training data is processed sentence by sentence. We used 3-word window representation for each ambiguous token including it, where neighbor tokens are 2 words from left. Each neighbor is represented by a vector of 126 morphological features. Since, a neighbor may have more than one parse; each vector is formed based on the existence of morphological features. Any morphological feature that exists in any parses of neighbor words is represented by 1. Illustrations for the problem generation phase is given in Fig. 1.

Table 3 Comparison of our proposed approach (using 10 different classifiers) with three different approaches

Method	Acc. (%)	Method	Acc. (%)
NaiveBayes	93.83	Logistic Regression	94.67
Conjunctive Rule	66.25	SVM	94.98
k-NN(k=10)	95.40	LWL	94.67
J48 Tree	**95.61**	Baseline Trigram-Based Model	**95.48**
J48 Tree(no prunning)	95.09	Greedy Prepend Algorithm	**95.82**
KStar	94.36	Perceptron(23 Features)	**96.28**
NNge	90.49		

Testing is done in a similar way as done in the training phase. The test set is divided into sentences. For each sentence, we select the tokens having more than one parses and form the instance vector using 3-word window. Then, we determine the corresponding problem. Using the model of the corresponding classifier we classify the test instance.

5 Experiments

5.1 Experimental Setup

We use a training set of approximately 1M semi-automatically tagged disambiguated tokens (including end-of-sentence, end-of-title, and end-of-document markers) taken from Turkish newspapers. The training data consists of 50673 sentences where about 40% of them are morphologically ambiguous [6]. The test set consists of 958 tokens including markers mentioned above, where 42 sentences and 379 tokens are morphologically ambiguous. Performance criterion is formulated as:

$$Performance = \frac{\#\ of\ correctly\ disambiguated\ tokens}{\#\ of\ tokens} \tag{5}$$

5.2 Results

We have compared our approach with other morphological disambiguation approaches for Turkish presented in Sect. 3 We used 10 different classification algorithms in the WEKA toolkit. The results of the classification algorithms and the previous approaches are given in Table 3. According to emprical results, J48 Tree classifier has the best performance among 10 classifier and Baseline Trigram-Based Model.

Although it cannot can attain the performance of the other previous approaches, the difference is not significant for the best performer namely J48 Tree classifier.

6 Conclusion

We presented a new approach to the morphological disambiguation problem for Turkish. Our proposed approach converts original morphological disambiguation problem into multiple classification problems. Each classification problem corresponds to a set of possible parses (not including the root words) where each parse maps to a class and correct parse map to the correct class for that instance. We used 10 different classifiers from WEKA toolkit for solving the classification problems.

Our experimental results show that the best classifier is J48 Tree classifier. Although this is only better than the Baseline Trigram-Based approach among the previous aproaches, we believe that by expanding our feature set and/or applying a linear/nonlinear feature extraction mechanism, we will achieve much better disambiguation performance.

References

1. Hakkani-Tür, D.Z., Oflazer, K., Tür, G.: Statistical morphological disambiguation for agglutinative languages. Comput. Humanit. **36**(4), 381–410 (2002)
2. Oflazer, K., Hakkani-Tür, D. Z., Tür, G.: Design for a Turkish treebank. In: Proceedings of the Workshop on Linguistically Interpreted Corpora (1999)
3. Sak, H., Güngör, T., Saraçlar, M.: Morphological disambiguation of Turkish text with perceptron algorithm. In: Gelbukh, A. (ed.) CICLING 2007, LNCS 4394, pp. 107–118 (2007)
4. Yüret, D., Türe, F.: Learning morphological disambiguation rules for Turkish. In: Proceedings of HLT-NAACL (2006)
5. Oflazer, K., Kuruöz, I.: Tagging and morphological disambiguation of Turkish text. In: Proceedings of the 4th Applied Natural Language Processing Conference, pp. 144–149 (1994)
6. Oflazer, K.: Two-level description of Turkish morphology. Lit. Linguist. Comput. **9**(2), 137–148 (1994)
7. Sak, H., Güngör, T., Saralar, M.: Turkish language resources: morphological parser, morphological disambiguator and web corpus. In: GoTAL 2008, volume 5221 of LNCS, pp. 417–427, Springer (2008)