A Hybrid Morphological Disambiguation System for Turkish

Mucahid Kutlu
Dept. of Computer Science & Eng.,
Ohio State University, Ohio, USA
kutlu@cse.ohio-state.edu

Ilyas Cicekli
Department of Computer Engineering,
Hacettepe University, Ankara, Turkey
ilyas@cs.hacettepe.edu.tr

Abstract

In this paper, we propose a morphological disambiguation method for Turkish, which is an agglutinative language. We use a hybrid method, which combines statistical information with handcrafted rules and learned rules. Five different steps are applied for disambiguation. In the first step, the most likely tags of words are selected. In the second step, we use handcrafted rules to constrain possible parses or select the correct parse. Next, the most likely tags are selected for still ambiguous words according to the suffixes of the words that are unseen in the training corpus. Then, we use transformation-based rules that are learned by a variation of Brill tagger. If the word is still ambiguous, we use some heuristics for the disambiguation. We constructed a hand-tagged dataset for training and applied a ten-fold cross validation with this dataset. We obtained 93.4% accuracy on the average when whole morphological parses are considered in calculation. The accuracy increased to 94.1% when only part-of-speech tags and inflections of last derivations are considered. Our accuracy is 96.9% in terms of part-of-speech tagging.

1 Introduction

Digital text sources increase every day and people can reach digital sources via Internet. The automatic processing of these sources becomes crucial in order to use and manage them effectively. Many NLP researchers work on different topics like summarization of texts, translation between natural languages, information extraction, etc. However, working on natural languages has difficulties due to their ambiguous natures. Since constraining possible morphological parses of words with disambiguation methods reduces the ambiguity problem, morphological disambiguation is crucial for performing better operations on texts.

In this paper, we propose a method for morphological disambiguation of Turkish texts. Turkish is an agglutinative language. Agglutinative languages generate words by joining affixes together and each affix represents one unit of meaning. This property loads many meanings to a single word. Since suffixes can increase the ambiguity by generating totally different words, the morphological analysis of words with many suffixes is much harder.

Our hybrid system uses statistical knowledge, learned transformation-based rules, handcrafted rules and some heuristics in the disambiguation process. In our system, we first obtain the statistical information about words and suffixes like frequencies of their corresponding morphological parses and learn transformation-based rules. In disambiguation, we use an iterative approach that applies techniques from the most reliable technique to the less reliable technique. First, if the word exists in our training set, we select the morphological parse with the highest frequency. Then, we consider handcrafted rules of SupervisedTagger software (Daybelge and Cicekli, 2007) for disambiguation. Next, we use statistical information about suffixes and select the morphological parse with the highest frequency in the corpus for that suffix. Then, we apply the disambiguation rules learned from the corpus. Finally, we use some heuristics, which depend on some statistical information to select morphological parses for still ambiguous words.

The major contribution of this study is our hybrid morphological system, which combines the statistical, and rule based approaches for disambiguation. The accuracy of our hybrid disambiguation system is quite good when examined with the Turkish language which has a very flexible set of grammar rules and very ambiguous words. Our combined approach works very well even with less statistical language sources such as a small hand-tagged corpus. Although the size of our hand-tagged corpus is relatively small, the performance of our hybrid system is very good. The performance of the presented hybrid system can be improved further when a bigger hand-tagged corpus is available.

The rest of the paper consists of four sections. Section 2 describes the related work in morphological disambiguation. Section 3 explains the proposed system. Section 4 describes the corpus and presents the performance results of the system. Section 5 concludes the paper by summarizing the study and discusses some future work.

2 Related Work

The related work about morphological disambiguation can be divided into three categories: statistical, rule based and hybrid which is the combination of the two approaches. Statistical approaches select the morphological parses using a probabilistic model that is built with the training set consisting of unambiguously tagged texts. There are various models used in the literature, such as, maximum entropy models (Ratnaparkhi, 1996; Toutanova and Manning, 2000), Markov Model (Church, 1988) and hidden Markov Model (Cutting et. al., 1992). In rule based methods, hand crafted rules are applied in order to eliminate some incorrect morphological parses or select correct parses (Daybelge and Cicekli, 2007; Oflazer and Tür, 1997; Voutilainen, 1995). These rules can also be learned from a training set using a transformation based (Brill, 1995) or memory based (Daelemans, 1996) learning approaches. There are also studies that combine statistical knowledge and rule based approaches (Leech et al., 1994; Tapanainen and Voutilainen, 1994; Oflazer and Tür, 1996).

The disambiguation studies can also be divided according to the languages they are applied. Levinger et al. (1995) used morpho-lexical probabilities learned from an untagged corpus for morphological disambiguation of Hebrew texts. Hajic and Hladká (1998) used maximum entropy modeling for Czech, which is an inflectional language. Morphological disambiguation of agglutinative languages, such as Turkish, Hungarian, Basque, etc., is harder than others because they have more morphological parses per words. Megyesi (1999) has used Brill's POS tagger with extended lexical templates to Hungarian. Hajic (2000) extended his work for Czech to five other languages including Hungarian. Ezeiza et al. (1998) combined statistical and rule based disambiguation methods for Basque. Rule based methods (Oflazer and Tür, 1997; Daybelge and Cicekli, 2007) and trigram-based statistical model (Tür et al., 2002) are used for the disambiguation of Turkish words. Yüret and Türe (2006) propose a decision list induction algorithm for learning morphological disambiguation rules for Turkish. Sak et al. (2007) apply perception algorithm in disambiguation of Turkish Texts.

3 Disambiguation System

A Turkish word can have many morphological parses containing many morphemes that give us morphological information about the word. For example, the word "çiçekçi" (florist) has the following ***morphological parse*** (MP)[1]:

```
çiçek+Noun+A3sg+Pnon+Nom
   ^DB+Noun+Agt+A3sg+Pnon+Nom.    (1)
```

The first part gives us the stem, which is "çiçek" (flower). We define the rest of the parse as the ***whole tag*** of the word. In parse, "`^DB`" shows that the word is derived from one type to another and its meaning has changed rather than its inflection. We define the final morphemes after the last derivation as the ***final tag*** of the word. For this example, the whole tag is:

```
Noun+A3sg+Pnon+Nom
   ^DB+Noun+Agt+A3sg+Pnon+Nom,      (2)
```

and the final tag is:

```
Noun+A3sg+Pnon+Nom
```

where the type of derivation "`Agt`" is ignored. The rules that are learned by our system depend on morphological parses, whole tags or final tags of words.

Our disambiguation system consists of two main parts: *training* and *disambiguation*. The training corpus is used for the induction of disambiguation rules and the generation of the tables *Most Likely Tag of Word Table* (WordTbl) and *Most Likely Tag of Suffix Table* (SuffixTbl). WordTbl is used to retag the corpus by our Brill tagger in order to learn rules. WordTbl, SuffixTbl and the learned rules are used in the disambiguation process.

The first table (WordTbl) holds frequencies of all morphological parses of words, and the second one (SuffixTbl) holds the frequencies of all possible morphological parses for suffixes. Of course, WordTbl also indicates most likely morphological parses of words since the highest frequency morphological parse is the most likely parse. Since all possible Turkish words cannot be seen in a training corpus, WordTbl will not hold most likely parses for all words. In order to make

[1] Noun is a major word category; A3sg is a noun agreement marker; Pnon is a noun possessive marker; Nom is a noun case marker; derivational boundaries are marked with ^DB.

an intelligent guess for the most likely parse of an unseen word, we use its suffix. For this purpose, we create SuffixTbl. In order to create SuffixTbl, we find suffixes of words according to their correct morphological parse and calculate the frequencies for tags corresponding to suffixes. For example, the suffix of the word "çiçekçi" (florist) whose morphological parse is given in (1) is "çi" and its corresponding whole tag is given in (2). We find frequencies of all corresponding whole tags for suffixes to store them in SuffixTbl.

3.1 Learning disambiguation rules

In order to learn disambiguation rules, we use a variation of Brill tagger. After all words in the training corpus are initially tagged with their most likely parses using WordTbl, disambiguation rules are learned. The learned disambiguation rules are based on morphological parses, whole tags, or final tags. The general format of a disambiguation rule is as follows:

if conditions ***then***
  select MPs containing TAG for $word_i$

The conditions of a rule depend on the possible MPs of the target word $word_i$ and the current selected MPs of the previous (or following) one or two words. Thus, the conditions of a rule can be one of the following:

- $wordC_i$ ***and*** $wordC_{i-1}$
- $wordC_i$ ***and*** $wordC_{i-1}$ ***and*** $wordC_{i-2}$
- $wordC_i$ ***and*** $wordC_{i+1}$
- $wordC_i$ ***and*** $wordC_{i+1}$ ***and*** $wordC_{i+2}$

Each condition $wordC_k$ is in the following form:

TAG of $word_k$ = TAG_a

TAGs appearing in the conditions or the MP selection part of a rule can be MPs, whole tags, or final tags.

In the learning of disambiguation rules, a variation of Brill tagger (Brill, 1995) is used. All possible rules are tried in order to select the rule that gives the best improvement. After applying the selected rule, we repeat the process in order to infer the other rules. These iterations end if there is no progress or the improvement is below a threshold. In the selection of the best rule, our method differs from the original Brill tagger. We select the rule with the highest precision as the best rule in iterations. For example, if rule A causes 100 correct tags and 1 wrong tag and rule B causes only 10 correct tags without any wrong tags, the original Brill tagger may choose the rule A for that iteration. However, we select rule B because it causes no mistakes. The reason for this approach is that we want to increase the correctness of the condition words in the rule applications for later steps of the algorithm, and mistakes in early stages can cause more mistakes in further steps.

The rules are learned using the dataset of 25098 hand-tagged words. After tagging all words in the training set with their most likely tags, we infer the best rule at each iteration step of the algorithm. We generate all possible rules from all the words in the dataset. After generating all rules, we select the rule with the highest precision as the best rule. If there is more than one rule with the highest precision, we select the one, which affects more words. When there is more than one rule with the highest precision and they affect the same number of words, we select any one of them. We applied ten-fold cross-validation for experiments. In the training part of the experiments, we have learned 395.2 rules on average.

3.2 Morphological Disambiguation

In the morphological disambiguation of Turkish words, we have used a hybrid disambiguation system, which uses statistical techniques, rule based techniques and some heuristics. After the given Turkish text is morphologically analyzed by a Turkish morphological analyzer, the hybrid disambiguation steps are applied.

In the morphological analysis of a given Turkish text, SupervisedTagger software (Daybelge and Cicekli, 2007) which uses a PC-Kimmo based morphological analyzer (Istek and Cicekli, 2007) is used. SupervisedTagger contains a morphological analyzer and a rule-based morphological disambiguation tool. The morphological parsing capability of SupervisedTagger is improved by using an updated unknown word recognizer and new heuristics for proper nouns and foreign words. If a word begins with a capital letter and it is not the first word of the sentence, it is assumed that it has also a proper name morphological parse even though it is not in the proper name list. In addition, the words that are not correct according to the Turkish grammatical rules are assumed to be foreign words that have proper name morphological parse. By these extensions, the average number of morphological parse per word is increased from 1.8 to 2.0.

In our hybrid disambiguation tool, we use the statistical information in tables WordTbl and SuffixTbl, hand-crafted rules of Supervised-Tagger, rules learned by our Brill tagger and some fall-back heuristics. The disambiguation algorithm consists of five major components:

- *Selection of the Most Likely Tag of Word*
- *SupervisedTagger Disambiguation*
- *Selection of the Most Likely Tag of Suffix*
- *Application of the Learned Rules*
- *Selection with Fall-Back Heuristics*.

The system tries to find the correct morphological parses step by step using the components in the given order. Correct parses of words can be selected or ambiguity levels of words can be reduced by eliminating some illegal parses. But words can be still ambiguous after intermediate steps. After the last step *Selection with Fall-Back Heuristics*, a single morphological parse will be definitely selected for each word.

Selection of the Most Likely Tag of Word (MW) - The statistical information in WordTbl helps us to find the most likely parses of words appearing in the training set. If the word exists in WordTbl, the most frequent parse is selected for that word. Since not all words appear in the training set, some words will be still ambiguous at the end of this step. WordTbl may not contain all words because our training data set is small, and the number of unique Turkish words is huge. In one of our experiments, we determined that the number of unique words is 870,000 in a 6 billion word Turkish corpus. In fact, this is one of the reasons that we decided to use a hybrid approach for the morphological disambiguation.

SupervisedTagger Disambiguation (ST) – In this step, the words are tried to be disambiguated by SupervisedTagger software. SupervisedTagger uses 342 hand-coded disambiguation rules of two types: *selection* and *elimination* rules. The selection rules select a morphological parse directly. The elimination rules eliminate the wrong ones as much as it can. In other words the selection rules completely disambiguate words, and the elimination rules reduce the ambiguity levels of words. SupervisedTagger is applied only to ambiguous words. At the end of this step, there can still be ambiguous words but the ambiguity level can be reduced by the rules of SupervisedTagger.

Selection of the Most Likely Tag of Suffix (MS) – If the word is not disambiguated by the first two steps, we try to disambiguate using the statistical information in SuffixTbl. The possible suffixes of a word are determined according to its morphological parses, and the most likely morphological parse corresponding to those suffixes is selected if the suffixes appear in SuffixTbl. The word may not be disambiguated at this step because of the huge number of possible suffixes. In one of our experiments, we also determined that the number of unique suffixes is 40,000 in a 6 billion word Turkish corpus.

Application of the Learned Rules (LR) – It can be considered that tagging words according to their frequency is not correct. In this step, we are trying to correct our mistakes and handle special cases by applying the rules that are learned by our Brill tagger. The order of rule application is the order of learning. The condition part of a rule contains a condition depending on the target word of the rule, and one or two more conditions depending on other condition words. A rule can be applicable to a target word if all of the following conditions are satisfied:

- Its condition words are completely disambiguated and satisfy their conditions.
- The target word is disambiguated and satisfies its condition, or the target word is ambiguous and one of its still possible parses satisfies its condition.
- At least one of the parses of the target word contains the correct tag given in the selection part of the rule.

When a rule is applied, the target word can be completely disambiguated, or some of its parses are selected as its possible parses. If the target word contains only one morphological parse satisfying the correct tag, it is disambiguated; otherwise its parses satisfying the correct tag are selected as possible parses for the next step and others are eliminated. For example, the following rule is applicable under the given conditions:

if (final TAG of word$_i$ = *Adverb* **and**
whole TAG of word$_{i-1}$ = *Noun+A3sg+P3sg+Nom*)
then select MPs with whole tag *Adjective* for word$_i$

If the whole tag of the selected MP of word$_{i-1}$ is *Noun+A3sg+P3sg+Nom,* the final tag of at least one of possible MPs for word$_i$ is *Adverb*, and word$_i$ contains at least one MP having the whole tag *Adjective*, then MPs containing *Adjective* tag are selected for word$_i$.

Selection with Fall-Back Heuristics (SH) - At this last step, a small number of words can still be ambiguous. In this step, we perform the selection with fall-back heuristics in order to disambiguate the remaining ambiguous words. We have determined the following four heuristics and applied them in the given order. The application order is determined empirically.

a) Selection of Non-Derived (SND) – SND heuristic selects the parses containing no derivation suffixes since non-derived words are more common than derived words.
b) Selection of Proper Noun (SP) - SP heuristic selects the proper noun senses of the words if their possible parses contain proper noun senses.
c) Selection of Noun (SN) - SN heuristic selects the parses that their part of speech tags are noun.
d) Selection of Shortest (SS) - After applying all techniques and heuristics, if the word is still not disambiguated, we select the shortest parse in terms of the character length.

Number of words	25098
Number of distinct words	8493
Average number of parses per word	1.982
Number of words with single parse	12260
Maximum number of parses in one word	16
Number of distinct parses	17934
Number of distinct whole tags	2052
Number of distinct final tags	343
Number of proper nouns	3305
Number of non-proper nouns	9670
Number of derived words	4772

Table 1. Statistics of Data Corpus

4 Evaluation

We have constructed a data corpus consisting of 25098 hand-tagged words. In the preparation of the corpus, we used Turkish texts from different news portals. Ten graduate students tagged words with correct morphological parses using SupervisedTagger software. The statistical information about our dataset is given in Table 1. There are 12 different part of speech tags which are noun, proper noun, conjunction, pronoun, adjective, question, interjection, verb, adverb, post-position, number and punctuation. The 48.8% of the corpus is unambiguous. The most ambiguous word has 16 different parses. There are 2052 distinct whole tags, which show the ambiguity problem of Turkish.

Our disambiguation system uses five different techniques (MW, ST, MS, LR, and SH) step by step. It is obvious that the order of the techniques is crucial for the performance of the system. In order to see which order gives the best accuracy, we have applied each technique separately and obtained the average accuracies by using 10 fold cross validation. In Table 2, the second column shows the average number of words having more than one parse and they are processed by the corresponding technique. The accuracy of the technique for the applied words is given in the third column. The fourth column shows the accuracy for disambiguated words so far (disambiguated words by the technique plus unambiguous words (UW)).

Technique	# of words applied	Acc. of Tech.	Acc. of (UW+Tech.)
MW	822.8	0.916	0.966
ST	802.4	0.798	0.919
MS	872.4	0.700	0.873
LR	26.6	0.744	0.994

Table 2. Results of techniques for the first step

Since MW gives the highest accuracy, it is reasonable to choose MW for the first step. Applying the learned rules at the first step is not reasonable since there are not enough disambiguated words yet. The reason for having high accuracy so far is that we have few words that are disambiguated in the first step, and unambiguous words in the corpus increase the accuracy.

Technique	# of words applied	Acc. of Tech.	Acc. of (UW+MW+Tech)
LR	18.9	0.741	0.967
ST	260.5	0.873	0.955
MS	369.4	0.761	0.934

Table 3. Results of techniques for the second step

For the second step, we tried MS, ST, and LR. The results are given in Table 3. When we compare Table 2 and Table 3, we can see that the accuracy of techniques increased, meaning that using more reliable techniques in the early steps causes an increase in the accuracy of other techniques by eliminating words that they cannot disambiguate correctly. Applying the learned rules (LR) at this step is again the worst technique. Using ST in the second step gives a higher accuracy than using MS. It is better to disambiguate more words in earlier steps with higher accuracy since the ambiguous words will be disambiguated with less reliable heuristics unless we disambiguate them at earlier steps. Thus, we select ST as the second, and MS as the third. Since ST and MS are better than LR, LR is chosen as the 4th component. The accuracy at the end of the 4th step is 0.942.

After the applications of the first four components, there are still some ambiguous words, and the number of ambiguous words after applying MW, ST, MS and LR is 71.4 on average (2.8%).

In order to disambiguate the remaining ambiguous words, we use fall-back heuristics. Since SND-SP-SN-SS order for the fall-back heuristics produced the best accuracy, we use that order. Finally, we disambiguated all words having the accuracy of 0.934 by using the order of MW-ST-MS-LR-SH.

SupervisedTagger uses handcrafted disambiguation rules. In order to measure the performance of the statistical components of our system, SupervisedTagger component is removed. The accuracy of the overall system is dropped from 0.934 to 0.924. This means that handcrafted rules help to improve the performance of the system. We believe that the importance of handcrafted rules will reduce significantly if we train our system with a huge tagged corpus.

In the calculation of the accuracy, we consider the whole morphological parse. However, in some words, all parses have same inflections after their last derivations so that they have the same grammatical function in the sentence. In other words, they have same final tags. In the calculation of the accuracy, if we consider only the final inflections (the final tags), the accuracy of the overall system becomes 0.941. When only the final part of speech tags are considered, the accuracy becomes 0.969.

Selection / **True**	Prop	Adj	Adv	Noun	Verb
Prop	6.6	5.7	1.2	12.3	2.4
Adj	1.3	4.2	2.0	6.0	1.5
Adverb	0.2	3.4	1.3	2.4	0.1
Noun	18.0	9.7	1.4	57.8	4.3
Verb	0.9	1.5	0.0	2.2	5.1

Table 4. The distribution of confusions

When we examined the errors of our system, we observed that most of the mistakes are in nouns, proper nouns, adjectives, verbs and adverbs. In Table 4, the distribution of wrong disambiguation is given. In the calculation, the average number of mistakes in every fold is used. We can see that adjectives are mostly confused with nouns. This is reasonable, since every adjective can also be used as noun in Turkish. Adverbs are also mostly confused with adjectives. Nouns are mostly confused with other nouns. This is an expected result since Turkish is an agglutinative language and there can be many different inflections from a stem. Verbs are most confused with verbs with different inflections. In addition, we can say that nouns are the POS tags mostly confused while adverbs are the least.

In our version of Brill Tagger we prefer the rules with minimum number of errors first instead of preferring the rules with most accuracy increase which the original Brill tagger uses. In order to see whether learning transformation based rules that cause no errors is useful or not, we defined a base method to apply our data set. In this base method, we select the most likely MP for a word from WordTbl if the word is in WordTbl. If it is not in WordTbl, the first MP for the word is selected. That is to say, the most likely MP for a word is selected randomly if it does not appear in the training corpus. Then we applied our learned rules and rules learned according to original Brill Tagger separately in order to see the difference between them. We again applied ten-fold cross validation and our variation had much higher accuracy than the original Brill Tagger.

5 Conclusion and Future Work

In this paper, we propose a hybrid disambiguation method that combines the statistical approaches with rule based approaches for Turkish. The first step is the selection of the most likely tags of words. If word is not disambiguated yet, we use hand-crafted rules. Then we use the most likely tags of suffixes for disambiguation. The learned transformation rules are applied in the fourth step.

For training and testing, we have constructed a relatively small corpus, which consists of highly ambiguous words. We applied ten-fold cross validation and obtained 93.4% accuracy on average when we considered whole morphological parses of words. The accuracy increases to 94.1% when final tags are considered. In addition, our accuracy is 96.9% for POS tags. When we use a huge corpus, we believe that our results will improve further.

We have used our components in different orders to see their effects. We observed that MW performs best. MS is better than ST for handling harder words. Considering ST together with the statistical approaches increases the performance of the system. The learned rules have also increased the accuracy.

Our system gives promising results in the disambiguation of Turkish words. Enlarging the corpus which will be useful for the statistical parts is left as a future work. In addition, examining different rule types and learning methodologies are also left for future work.

References

E. Brill, Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging, *Computational Linguistics*, 21(4):543–565, (1995).

K.W. Church, A stochastic parts program and noun phrase parser for unrestricted text, *Proceedings of the Second Conference on Applied Natural Language Processing*, pp:136-143, (1988).

D. Cutting, J. Kupiec, J. Pedersen and P. Sibun, A practical part-of-speech tagger, *Proceedings of the Third Conference on Applied Natural Language Processing*, pp:133-140, (1992).

W. Daelemans, J. Zavrel, P. Berck and S. Gillis, MBT: A memory-based part of speech tagger-generator, *Proceedings of the Fourth Workshop on Very Large Corpora*, pp:14-27, (1996).

T. Daybelge, and I. Cicekli, A Rule-Based Morphological Disambiguator for Turkish, *Proceedings of Recent Advances in Natural Language Processing*, pp:145-149, (2007).

N. Ezeiza, I. Alegria, J.M. Arriola, R. Urizar and I. Aduriz, Combining Stochastic and Rule based Methods for Disambiguation in Agglutinative Languages, *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pp:379-384, (1998).

J. Hajic and B. Hladká, Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset, *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pp:483-490, (1998).

J. Hajic, Morphological Tagging: Data vs. Dictionaries, *Proceedings of the Applied Natural Language Processing and the North American Chapter of the Association for Computational Linguistics*, (2000).

D.Z. Hakkani-Tür, K. Oflazer and G. Tür, Statistical Morphological Disambiguation for Agglutinative Languages, *Computers and the Humanities* 36(4), (2002).

O. Istek and I. Cicekli, A Link Grammar for an Agglutinative Language, *Proceedings of Recent Advances in Natural Language Processing*, pp:285-290, (2007).

G. Leech, R. Garside and M. Bryan, 1994. CLAWS4: The tagging of the British National Corpus, *Proceedings of COLING*, pp:622-628, (1994).

M. Levinger, U. Oman and A. Itai, Learning Morpho-Lexical Probabilities from an Untagged Corpus with an Application to Hebrew, *Computational Linguistics* 21(3), 383-404, (1995).

B. Megyesi, Improving Brill's POS Tagger for an Agglutinative Language, *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp:275-284, (1999).

K. Oflazer and G. Tür, Combining Hand-crafted Rules and Unsupervised Learning in Constraint-based Morphological Disambiguation, *Proceedings of the ACL-SIGDAT Conference on Empirical Methods in Natural Language Processing*, (1996).

K. Oflazer, and G. Tür, Morphological Disambiguation by Voting Constraints, *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, (1997).