# Turkish Treebank as a Gold Standard for Morphological Disambiguation and Its Influence on Parsing

**Özlem Çetinoğlu**

IMS, University of Stuttgart
Germany
ozlem@ims.uni-stuttgart.de

**Abstract**

So far predicted scenarios for Turkish dependency parsing have used a morphological disambiguator that is trained on the data distributed with the tool(Sak et al., 2008). Although models trained on this data have high accuracy scores on the test and development data of the same set, the accuracy drastically drops when the model is used in the preprocessing of Turkish Treebank parsing experiments. We propose to use the Turkish Treebank(Oflazer et al., 2003) as a morphological resource to overcome this problem and convert the treebank to the morphological disambiguator's format. The experimental results show that we achieve improvements in disambiguating the Turkish Treebank and the results also carry over to parsing. With the help of better morphological analysis, we present the best labelled dependency parsing scores to date on Turkish.

**Keywords:** Morphological Disambiguation, Parsing, Turkish

## 1. Introduction

Realistic statistical dependency parsing scenarios on morphologically rich languages (MRLs) require components that provide parsers with predicted lemmas, POS and morphological features. Turkish, as an MRL, also needs a segmentation component as words are segmented into sublexical units in the Turkish Treebank (Oflazer et al., 2003). The aim of the segmented representation is to explicitly represent word-internal relations that reflect morphosyntactic interactions (cf. Çetinoğlu and Kuhn (2013) for a discussion). The segmentation issue deviates Turkish from standard raw text parsing pipelines which has the general assumption of gold tokenisation. Instead, the segments of a word - together with its lemma, POS, and morphological features - are determined by morphological analysis.

Figure 1 depicts the morphological analysis of the word *tartışma* 'discussion'.[1] `^DB` denotes a derivational boundary, which segments the word into two sublexical units, namely *inflectional groups* (IGs hereafter). The first IG of a word provides its lemma. Each IG has its own POS and morphological features. IGs are the basic units in the Turkish Treebank representation, thus in parsing it.

So far only Eryiğit (2012) and Çetinoğlu and Kuhn (2013) experimented dependency parsing of Turkish in predicted settings. Both works use Oflazer's (1994) morphological analyser and Sak et al.'s (2008) disambiguator to retrieve automatic morphological analyses. The morphological analyser outputs all possible analyses of a word and the perceptron based disambiguator (**MD** henceforth) reduces the ambiguous analyses to one.

The data the disambiguator is trained on comes within the tool.[2] It is semi-automatically disambiguated, hence all training, development and test sets contain mistakes and unknowns. The semi-gold nature of the data obviously drops the accuracy of the disambiguator. The drop is substantial when the Turkish Treebank (**TTB** henceforth) is disambiguated, which then propogates to the parsing scores. In addition to the data accuracy problem, another reason for lower disambiguation scores on the treebank data is the domain difference between the MD and TTB data. The MD data contains newspaper text only whereas the TTB is built from a subset of the METU Corpus (Say et al., 2002), which includes sentences mainly from newspapers, essays, interviews, short stories, research monographs.

On the other hand, despite some annotatation errors and inconsistencies, the Turkish Treebank has the manually corrected gold features to train the disambiguator. Moreover, using the in-domain data can help improve the disambiguation of TTB inputs. The only obstacle to use the treebank as training data for the MD is the different formatting.

In this paper we utilise the gold morphological analyses from the TTB as our training data in the disambiguator with the help of some conversion scripts. We experiment using the converted data standalone and together with the existing MD data. Our experiments show the TTB data improves the morphological disambiguation and syntactic parsing of the Turkish Treebank and help achieve state-of-the-art labelled parsing scores. The converted TTB training data and the parser model are available to interested researchers.

## 2. Treebank as MD Training Data

To be able to use the TTB as MD training data we *i)* map unmatching tags *ii)* convert the format *iii)* combine the gold analysis with its alternatives in ambiguous cases.

TTB and MD representations are quite similar yet different. They use almost the same morphological tag set with minor differences given in Table 1. In addition to the mappings in the table, there are a couple of lexical exceptions. For instance, pronouns *hepsi* 'all', *bazısı* 'some', *biri* 'one'

[1] Pos: Positive, Inf2: Infinitive, A3sg: 3rd personal singular agreement, Pnon: no possessives, Nom: Nominative.

[2] http://www.cmpe.boun.edu.tr/~hasim/download/MD-Release.zip

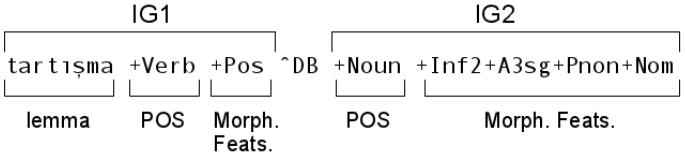


Figure 1: The morphological analysis of the word *tartışma* 'discussion'

| TTB Format | MD Format |
|---|---|
| +Adv | +Adverb |
| +Distrib | +Dist |
| +As | +AsLongAs |
| +AsLongAdamantly | +AsIf |
| +WithoutHavingDoneSo | +WithoutBeingAbleTo-HaveDoneSo |
| +{Demons\|Ques\|Pers\|Reflex}P | +{Demons\|Ques\|Pers\|Reflex} |
| +{A\|N}{Fut\|Past\|Pres}Part | +{Fut\|Past\|Pres}Part |
| +NInf | +Inf{1\|2\|3} |

Table 1: The differences between the Treebank and MD training data representation

have a `+Quant` tag in the MD representation although they are not categorised in the treebank. The conversion handles such cases too.

The second step of the transformation is to convert the format of the morphological features. In the TTB, a word is segmented into segments from its derivational boundaries (ˆDB). If a word is derived n times, it is represented as n+1 segments. The first segment has the lemma, and the last segment has the whole word as the surface form. The surface forms of non-final segments are underscores. Figure 2.a shows the TTB representation of *tartışma* 'discussion'. It is derived from the verb *tartış* 'discuss' with the infinitival suffix *-ma*. In the MD format, all segments are concatenated into a single morphological analysis.

The last step of the transformation is getting all morphological analyses of a word and sorting them. MD expects all the analysis of a word on the same line and the gold analysis as the first one among them. We retrieve the gold information from the TTB data and other analyses from Oflazer's (1994) morphological analyser. Figure 2.b displays an MD training data line. The word *tartışma* is ambiguous with a second analysis 'do not discuss'. The 'discussion' analysis precedes the second meaning.

## 3. Experimental Setup

### 3.1. Data Sets

The MD data contains 753K words in the training set (excluding sentence, title, and document boundaries). The development and test sets are 42K words each. There is also a manually corrected test set of 862 words. The METU-Sabancı Turkish Treebank (Oflazer et al., 2003) has a training set of 5635 sentences, which correspond to 56K words. The 300 sentences of ITU validation set (Eryiğit, 2007) is used for testing and has 3,7K words. There are no separate development sets. We use the *detached* version of the TTB (Eryiğit et al., 2011) where multiword expressions are represented as separate tokens.

### 3.2. Tools

In training data preparation, we implemented a set of small scripts for the TTB to MD conversion. In the parsing pipeline, after we morphologically analyse (Oflazer, 1994) and disambiguate (Sak et al., 2008) the parser input we assign analyses to unknown words based on some heuristic rules. As the dependency parser we adopt Bohnet's (2010) state-of-the-art graph-based parser.

### 3.3. Evaluation

The evaluation metric for morphological analysis is exact match. For parsing, we use an evaluation tool based on IGs (Eryiğit et al., 2008). The unlabelled attachement score $\text{UAS}_{IG}$ gives the ratio of IGs that are attached to the correct head, and the labelled attachement score $\text{LAS}_{IG}$ gives the ratio of IGs attached to the correct head with the correct label. In cases where the morphology (segmentation, POS, and morphological features) of the head word is different from the gold one, an attachement is correct only if the dependent is attached to the correct word *and* the head IG has the gold main POS. We omit punctuation in parser evaluation.

## 4. Experiments

We conduct two sets of experiments. The disambiguation experiments provide intrinsic evaluation where we disambiguate the development and test sets provided within the MD and also evaluate on the TTB training folds and test set. As an extrinsic evaluation we parse the TTB data sets disambiguated with different models and observe the effect of morphological features on parsing. We train the MD with four different data sets. The first data set is the original training data provided within the MD. The second set is the TTB training data converted to the MD training format. As a third experiment we combine the two data sets. The size of the MD training data is almost 15 times larger than the size of the TTB data. To make them equally weighted, we create a fourth training set with one copy of the MD data and 15 copies of the treebank.

The architectures that involve the TTB data require training 10 different models. Since the TTB does not have a separate development set, we apply 10-fold cross-validation on the training data in parsing experiments. We apply the same settings in disambiguation too, that is we split the TTB training data into 10 folds, use 9 folds as our MD training data and disambiguate the remaining fold with the trained model. The same 9 folds are used as training data in the parser too. This way, we ensure that the disambiguated and parsed fold is unseen data for the disambiguator and parser. The scores given are the averages of scores of 10 folds.

```
   ID  Form        Lemma    POS    Morph. Feat.
a. 1   _           tartış   Verb   Pos
   2   tartışma    _        Noun   Inf|A3sg|Pnon|Nom

b. tartışma tartış+Verb+Pos^DB+Noun+Inf2+A3sg+Pnon+Nom tartış+Verb+Neg+Imp+A2sg
```

Figure 2: The TTB and MD training representations of the ambiguous word *tartışma* 'discussion'

| Model | MD Dev | MD Test | Man. Test | 10-fold TTB Train Avg. | TTB Test |
|---|---|---|---|---|---|
| MD Train | **97.82** | **97.82** | **96.29** | 87.64 | 88.41 |
| TTB Train | 87.21 | 87.11 | 88.17 | **90.19** | **89.87** |
| MD Train + TTB Train | 97.79 | 97.76 | 95.94 | 88.26 | 88.74 |
| MD Train + 15 TTB Train | 97.62 | 97.51 | 96.17 | 89.06 | 89.09 |

Table 2: **Morphological disambiguation scores.** Man. Test corresponds to the MD manually corrected test set and 10-fold TTB Train Avg. corresponds to the 10-fold cross validation average scores on the TTB training set.

In disambiguating and parsing the TTB test data, we use the whole training data in training models.

Both for intrinsic and extrinsic evaluation we compare the results of MD Train, the model used in parsing pipelines in previous work, and TTB Train, our best model.

### 4.1. Intrinsic Evaluation: Disambiguation Experiments

Table 2 gives the morphological accuracy of four models on two groups of data sets. In the MD group, using only the MD training data has already high scores. When only the TTB data is used, there is a huge drop up to 10 points on all three data sets. Having such a drop despite the gold TTB annotations indicates the domain difference. When both of the data sets are used in training, automatic and manual test sets recover their accuracy. Adding more TTB data slightly harms the accuracy.

In the TTB group the trend completely changes. Using the original MD training data does not give as accurate scores on the TTB sets as on the MD sets. This observation complies with the morphological evaluation scores given in (Eryiğit, 2012) where she observed almost 9 point difference between the reported accuracy in (Sak et al., 2008) and the TTB training and test sets. When the TTB data is used there is a 2.5% absolute jump in the training set 10-fold cross validation and and a 1.5% absolute jump in the test set results. Combining both data sets improves over using only the MD data, but does not outperform using only the TTB data.

### 4.2. Disambiguation Error Analysis

In order to investigate the advantage of using the TTB data over the MD data, we compare the two systems MD Train and TTB train on the cross-validated TTB training set. Table 3 shows the error breakdown of two systems according to the number of IGs in an analysis, hence gives us insight about segmentation. The second row of the table shows that for most of the incorrect analyses, the segmentation is still correct. Both models tend to undersegment (PredIG < GoldIG) rather than oversegment (PredIG > GoldIG) when they make a segmentation mistake.

The distribution of the errors show that when we move from the MD Train model to the TTB Train model, the rate of the equally segmented errors drop the most (2.28%). Among those incorrect analyses with correct segmentation, the MD Train model can find 77.07% of the lemmas and 48.23% of the POS correctly. These numbers increase to 78.75% and 50.44% respectively when the data is disambiguated by the TTB Train model.

| Predicted vs. Gold | MD Train | TTB Train |
|---|---|---|
| Exact Match | 49449 (87.64%) | 50891(90.19%) |
| Err: PredIG = GoldIG | 5260 (9.32%) | 3971 (7.04%) |
| Err: PredIG < GoldIG | 1283 (2.27%) | 1176 (2.08%) |
| Err: PredIG > GoldIG | 432 (0.77%) | 368 (0.68%) |

Table 3: Comparison of the number of IGs when the predicted and gold morphological analyses mismatch. PredIG and goldIG denote the number of IGs in a predicted and a gold analysis respectively.

Table 4 gives the 10 most frequent disambiguation mistakes the MD Train model made on the cross-validated TTB training data. For comparison, the table also shows the most frequent TTB Train model mistake for each of the 10 words.

When the MD Train model is used the most frequent mistake is to assign *ile* the postposition meaning 'with' instead of the conjunction meaning 'and'. When the disambiguator is trained on the TTB data the frequency of this mistake drastically drops to 4. However, another mistake emerges with 15 occurrences; this time the postposition meaning is incorrectly identified as a conjunction. The disambiguation choices by two models directly reflect the distribution of *ile* in their training data. In the MD data the gold representation for *ile* has dominantly the postposition sense which leads the disambiguation model to learn to pick that sense more. In the TTB data, on the other hand, the gold analysis is mostly the conjunction sense and when disambiguates, the TTB Train system takes advantage of being trained on the in-domain data.

*Bir* is a frequent and quite ambiguous word in Turkish, meaning the determiner 'a', the number 'one', the adjectives 'one/same' and the adverbs 'once/only'. It is no surprise that both models find it hard to disambiguate it correctly. In the table, only the most frequent predicted-gold

| Word | Model | Predicted | Gold | Count |
|---|---|---|---|---|
| *ile* 'with/and' | **MD:** | ile+Postp+PCNom | ile+Conj | 91 |
| | **TTB:** | ile+Conj | ile+Postp+PCNom | 15 |
| *bir* 'a/one' | **MD:** | bir+Det | bir+Num+Card | 88 |
| | **TTB:** | bir+Det | bir+Num+Card | 76 |
| *daha* 'more' | **MD:** | daha+Noun+A3sg+Pnon+Nom | daha+Adverb | 79 |
| | **TTB:** | daha+Noun+A3sg+Pnon+Nom | daha+Adverb | 1 |
| *değil* 'not' | **MD:** | değil+Conj | değil+Verb+Pres+A3sg | 63 |
| | **TTB:** | değil+Conj | değil+Verb+Pres+A3sg | 63 |
| *var* 'existent' | **MD:** | var+Adj | var+Adj^DB+Verb+Zero+A3sg | 56 |
| | **TTB:** | var+Adj | var+Adj^DB+Verb+Zero+A3sg | 51 |
| *bu* 'this' | **MD:** | bu+Det | bu+Pron+Demons+A3sg+Pnon+Nom | 51 |
| | **TTB:** | bu+Det | bu+Pron+Demons+A3sg+Pnon+Nom | 43 |
| *o* 'that' | **MD:** | o+Det | o+Pron+Pers+A3sg+Pnon+Nom | 46 |
| | **TTB:** | o+Det | o+Pron+Pers+A3sg+Pnon+Nom | 22 |
| *böyle* 'such/so' | **MD:** | böyle+Adj | böyle+Adverb | 46 |
| | **TTB:** | böyle+Adverb | böyle+Adj | 13 |
| *ne* 'which/what/neither' | **MD:** | ne+Adj | ne+Pron+Ques+A3sg+Pnon+Nom | 46 |
| | **TTB:** | ne+Pron+Ques+A3sg+Pnon+Nom | ne+Conj | 10 |
| *onu* 'that+Acc' | **MD:** | o+Pron+Demons+A3sg+Pnon+Acc | o+Pron+Pers+A3sg+Pnon+Acc | 42 |
| | **TTB:** | o+Pron+Pers+A3sg+Pnon+Acc | o+Pron+Demons+A3sg+Pnon+Acc | 7 |

Table 4: The 10 most frequent disambiguation mistakes the MD Train model made on the cross-validated TTB training data. For each word, the first row gives the MD Train model and gold analyses. The second row gives the most frequent disambiguation mistake made by the TTB Train model for the same word.

mismatch is shown for *bir* but determiner vs. adverb and determiner vs. adjective mismatches are also frequent for this word.

Oflazer's morphological analyser gives two outputs for the adverb *daha* 'more/still/yet'. One is a frequent adverb analysis, and the other one is a noun analysis. This noun analysis never occurs in the MD training data. Moreover when there are both a noun and an adverb analysis of a word, the noun analysis is the gold one most of the time. As a consequence the MD Train model heavily fails to assign the correct adverb analysis to *daha*. In the TTB Train model, the training set combines the gold TTB analyses with the morphological analyser output. Hence the model sees several times the correct choice among two possibilites during training. During disambiguation, the TTB Train model makes a mistake only once.

Table 4 continues with errors that are highly frequent in both models. The conjunction sense of *değil* corresponds to 'not' in the phrase *Ali değil Ahmet* 'not Ali but Ahmet'. Its verb sense is used in copular negation. Both senses exist in the training sets, it's a comparatively harder classification problem for the disambiguator. *Var* 'existent' is a frequently used adjective due to its role in copular sentences, e.g. *Bir kedi var.* 'There is a cat.' (lit. A cat existent). The gold analysis in the table represents the copular use of *var*. Both data sets are noisy in marking the gold analysis among possible alternatives. It is expected to have such noise in the MD Train data due to its semi-automatic nature, but for *var* inconsistencies are frequent in the TTB too.

Both *bu* and *o* are frequent words with several meanings. *Bu* is the determiner and demonstrative pronoun 'this', and similarly *o* is the determiner and demonstrative pronoun 'that' and the personal pronoun 'he/she/it'. Both systems confuse the pronoun meanings with the determiner meaning. Among less frequent mistakes, other combinations of mismatches are also observed. *Böyle* means the adjective 'such' and the adverb 'so'. In the MD data, the adjective sense is marked as gold more frequently than the adverb sense, hence the disambiguator has a tendency to pick the adjective meaning more often, which leads to mistakes. Such mistakes disappear when the TTB data is used, but this time the disambiguator is biased towards the adverb sense.

*Ne* is yet another frequent and highly ambiguous word. It has the adjective meaning 'which', the interrogative pronoun meaning 'what', and the conjunction meaning 'neither, nor' in constructions like *ne Ali ne Ahmet* 'neither Ali nor Ahmet' together with adverbial and interjetion meanings. The MD Train model selects the adjective meaning instead of the interrogative pronoun meaning 46 times. In the TTB Train model this drops down to 2 instances. But it incorrectly assigns the interrogative sense to *ne* conjunctions 10 times. The last word in Table 4 is the accusative form of *o* 'that/he/she/it'. The MD Train model confuses the personal pronoun with the demonstrative pronoun and using the TTB training data solves this problem. There is still some confusion in the TTB Train model, this time with a bias towards the demonstrative pronoun.

### 4.3. Extrinsic Evaluation: Parsing Experiments

In the second set of experiments we use the TTB files disambiguated by different models as input to a dependency parser, and observe if improvements in morphological features propagate in parsing.

Table 5 gives the labelled and unlabelled accuracy scores on the cross-validated training set and test set. Models correspond to the different data sets used in training the disambiguator. The TTB data sets pattern we observe in Table

| | 10-fold Train Avg. | | Test | |
|---|---|---|---|---|
| **Model** | **LAS$_{IG}$** | **UAS$_{IG}$** | **LAS$_{IG}$** | **UAS$_{IG}$** |
| MD Train | 61.82 | 73.89 | 64.40 | 76.04 |
| TTB Train | **63.53** | **75.24** | **65.38** | **76.89** |
| MD + TTB | 62.12 | 74.19 | 64.56 | 76.29 |
| MD + 15 TTB | 62.48 | 74.36 | 64.50 | 76.35 |
| Ç&K 2013 | 62.58 | 74.35 | 65.19 | 77.05 |

Table 5: **TTB parsing scores.** 10-fold Train Avg. corresponds to the 10-fold cross validation average scores on the TTB training set. Ç&K 2013 gives the best parsing scores from (Çetinoğlu and Kuhn, 2013).

2 follows in this table too. The MD Train has the lowest scores. Using the MD and TTB data together improves the scores a bit, and there is a slight improvement on top of that if more copies of the TTB are used. Both for the 10-fold cross validation on training data and test data, the best scores are achieved when the most accurate morphological features are used, that is, the disambiguator is trained on the TTB data only.

We compare our best parsing scores with the best published scores from (Çetinoğlu and Kuhn, 2013). On the cross-validated training data the parser that uses the TTB as the disambiguator training data outperforms the best system from that work, which uses a joint parser (Bohnet and Nivre, 2012) and a model that takes advantage of the correlation between case markers and argument relations. Better score also holds for the test set in labelled attachement score although the difference is less prominent. And the Ç&K 2013 is slightly better in the unlabelled attachement score.[3]

### 4.4. Parsing Error Analysis

In Table 6 we give the dependency breakdown of two systems and their precision recall scores. The MD Train and TB Train systems correspond to the first and second rows of Table 5 respectively. The breakdown show that except for question particle and vocative recall values, all dependencies increased in their precision and recall when the TTB is used as training data in the preprocessing step of the parser. The highest improvements are in subjects, multiword expressions, instrumental adjuncts, negative particles, appositions, ablative adjuncts, determiners, classifiers, and intensifiers.

We also look into a treebank sentence to exemplify how better disambiguation accuracy helps improve the parsing accuracy. Example 1 is composed of three ambiguous words. *Haftalar* which normally means 'weeks' also includes an improbable verb meaning. *sonra* has the postposition meaning 'after', the adverb meaning 'later' and the noun meaning 'after'. The verb *bulmuş* 'found' is also an adjectival participle.

[3]Note that better preprocessing and utilising case markers are orthagonal solutions to the parsing problem. It is possible to try the latter method on top of our best system from this work and test if we can achieve even higher parsing scores. Our main concern in this work is to observe the impact of better morphological disambiguation, thus we leave this option aside.

| | MD Train | | TTB Train | |
|---|---|---|---|---|
| **Dependency** | **Prec.** | **Rec.** | **Prec.** | **Rec.** |
| SUBJECT | 46.6 | 48.9 | 49.4 | 50.4 |
| S.MODIFIER | 52.5 | 45.1 | 53.1 | 46.1 |
| MWE | 63.8 | 59.5 | 66.2 | 60.0 |
| QUESTION.PARTICLE | 66.7 | 60.4 | 62.8 | 64.6 |
| INSTRUMENTAL.ADJUNCT | 22.6 | 17.3 | 28.7 | 22.1 |
| DATIVE.ADJUNCT | 41.6 | 43.5 | 43.4 | 45.5 |
| NEGATIVE.PARTICLE | 63.9 | 58.8 | 66.7 | 58.8 |
| COORDINATION | 49.5 | 46.2 | 50.5 | 51.7 |
| OBJECT | 57.4 | 57.6 | 59.6 | 58.7 |
| SENTENCE | 87.5 | 88.5 | 87.9 | 88.9 |
| APPOSITION | 31.6 | 12.8 | 35.1 | 13.9 |
| LOCATIVE.ADJUNCT | 42.2 | 44.2 | 42.7 | 44.6 |
| VOCATIVE | 36.0 | 20.7 | 35.8 | 24.1 |
| ABLATIVE.ADJUNCT | 39.9 | 41.5 | 42.1 | 43.2 |
| DETERMINER | 69.6 | 84.0 | 72.1 | 84.2 |
| CLASSIFIER | 59.4 | 70.2 | 63.3 | 68.9 |
| MODIFIER | 59.5 | 56.7 | 60.6 | 60.2 |
| INTENSIFIER | 68.8 | 67.6 | 73.9 | 73.9 |
| POSSESSOR | 71.7 | 72.7 | 72.2 | 73.9 |

Table 6: The dependency breakdown of the cross-validated training data scores. MD Train and TTB Train denote the parsing input disambiguated by the MD Train and TTB Train models respectively.Precision and recall are given in percent. Dependencies with less than 100 occurrences are omitted.

(1) Hafta-lar sonra bul-muş.
Week-Pl after find-Narr.A3sg
'S/he has found (it) after weeks.'

Figure 3 gives the predicted morphological features and dependency trees of the example sentence 1. When the sentence is disambiguated by the MD Train model (left), it correctly identifies the morphological analyses of *haftalar* and *bulmuş*. However the word *sonra* incorrectly has the adverb analysis. During parsing the parser decides the adverb modifies the verb, and the noun in nominative case is the subject. Both decisions make sense given the morphological analyses and Turkish grammar rules, yet the subject dependency is not correct.

When the disambiguator is trained on the TTB data, it manages to correctly assign the postposition meaning to *sonra*.

## 5. Conclusion

Our results confirm better preprocessing (segmentation, lemmatisation, POS and morphological feature tagging) lead to better parsing for MRLs (Björkelund et al., 2013). Another conclusion we derive from our experiments is that in the MD system a model trained on a smaller yet more accurate data set outperforms a model trained on a larger less accurate data set.

According to our results, using the same training data in morphological disambiguation and parsing improves parsing accuracy. We also outperform the existing scores for Turkish dependency parsing with the help of more accurate preprocessing. Only changing to a better predicted input causes an almost 1 point LAS$_{IG}$ gain over the baseline sys-

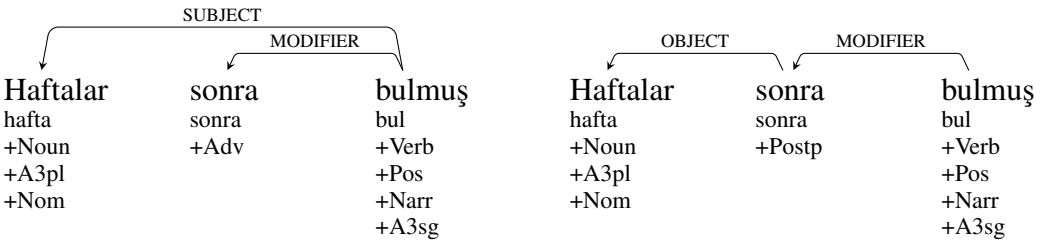


Figure 3: The example sentence disambiguated by the MD Train model (left) and the TTB Train model (right), and their dependency trees.

tem on the TTB test set by Bohnet's (2010) graph-based parser.

We provide the interested researchers with the TTB training data in the MD training data format and the parser model trained on the TTB training data on the following webpage: http://www.ims.uni-stuttgart.de/ ozlem/cetinogluLREC14.html

## 6. Acknowledgements

We thank Fatih Dağaşan for his contributions to the conversion scripts. This work is funded by the Collaborative Research Centre (SFB 732) at the University of Stuttgart.

## 7. References

Anders Björkelund, Özlem Çetinoğlu, Richárd Farkas, Thomas Mueller, and Wolfgang Seeker. 2013. (re)ranking meets morphosyntax: State-of-the-art results from the SPMRL 2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 135–145, Seattle, Washington, USA, October. Association for Computational Linguistics.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the EMNLP-CoNLL*, pages 1455–1465, Jeju, Korea.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of COLING*, pages 89–97, Beijing, China.

Özlem Çetinoğlu and Jonas Kuhn. 2013. Towards joint morphological analysis and dependency parsing of Turkish. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 23–32, Prague, Czech Republic, August. Charles University in Prague, Matfyzpress, Prague, Czech Republic.

Gülşen Eryiğit. 2007. ITU validation set for METU-Sabancı Turkish treebank.

Gülşen Eryiğit. 2012. The impact of automatic morphological analysis & disambiguation on dependency parsing of Turkish. In *Proceedings of LREC*, Istanbul, Turkey.

Gülşen Eryiğit, Joakim Nivre, and Kemal Oflazer. 2008. Dependency parsing of Turkish. *Computational Linguistics*, 34(3):357–389.

Gülşen Eryiğit, Tugay Ilbay, and Ozan Arkan Can. 2011. Multiword expressions in statistical dependency parsing. In *Proceedings of the SPMRL Workshop of IWPT*, pages 45–55, Dublin, Ireland.

Kemal Oflazer, Bilge Say, Dilek Zeynep Hakkani-Tür, and Gökhan Tür. 2003. Building a Turkish treebank. In Anne Abeille, editor, *Building and Exploiting Syntactically-annotated Corpora.* Kluwer Academic Publishers, Dordrecht.

Kemal Oflazer. 1994. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 9(2):137–148.

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2008. Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In *Proceedings of GoTAL 2008*, pages 417–427.

Bilge Say, Deniz Zeyrek, Kemal Oflazer, and Umut Özge. 2002. Development of a corpus and a treebank for present-day written Turkish. In *Proceedings of the 11th International Conference on Turkish Linguistics*.