ITU Treebank Annotation Tool

Gülşen Eryiğit
Department of Computer Engineering
Istanbul Technical University
Istanbul, 34469, Turkey
gulsen.cebiroglu@itu.edu.tr

Abstract

In this paper, we present a treebank annotation tool developed for processing Turkish sentences. The tool consists of three different annotation stages; morphological analysis, morphological disambiguation and syntax analysis. Each of these stages are integrated with existing analyzers in order to guide human annotators. Our semi-automatic treebank annotation tool is currently used both for creating new data sets and correcting the existing Turkish treebank.

1 Introduction

Annotated corpora is essential for most of the natural language processing tasks. Developing new annotated corpora becomes crucial especially for lesser studied languages where we encounter many difficulties for finding such data. Turkish is one of the languages which still suffer from scarcity of annotated resources. The most reliable data set for Turkish is the Metu-Sabancı Turkish Treebank (Oflazer et al., 2003) consisting of 5635 sentences annotated with dependency structures. Unfortunately, the data size of this treebank remained unchanged during recent years. There exist also some other small data sets manually pos-tagged by different research groups.

In this study, we introduce our treebank annotation tool developed in order to improve the size of the existing data sets for Turkish (particularly the treebank). Our main motivation for developing a new tool is the inability of the existing tools (e.g. Atalay et al. (2003) and DepAnn (Kakkonen, 2006) which seems to be the most suitable tools for our task) in either reflecting the peculiar morphological and dependency structure of Turkish or providing suitable automatic analyses for guidance. We also aim to speed up the annotation process by using graphical user-friendly interfaces and transforming the annotation process from a manual (starting from scratch) procedure into a controlling and correcting procedure. In the rest of this paper, we first introduce the framework of the tool and then the details of its different annotation stages. We then close with conclusions and future work.

2 Framework

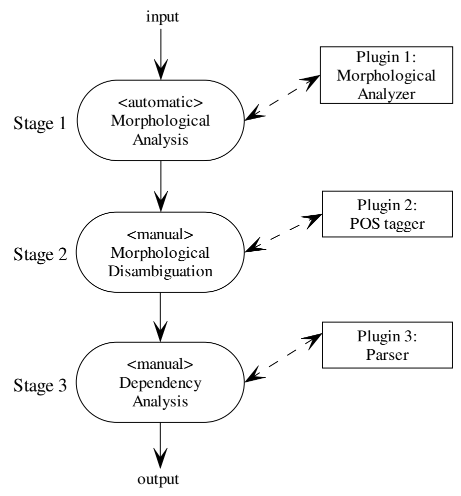
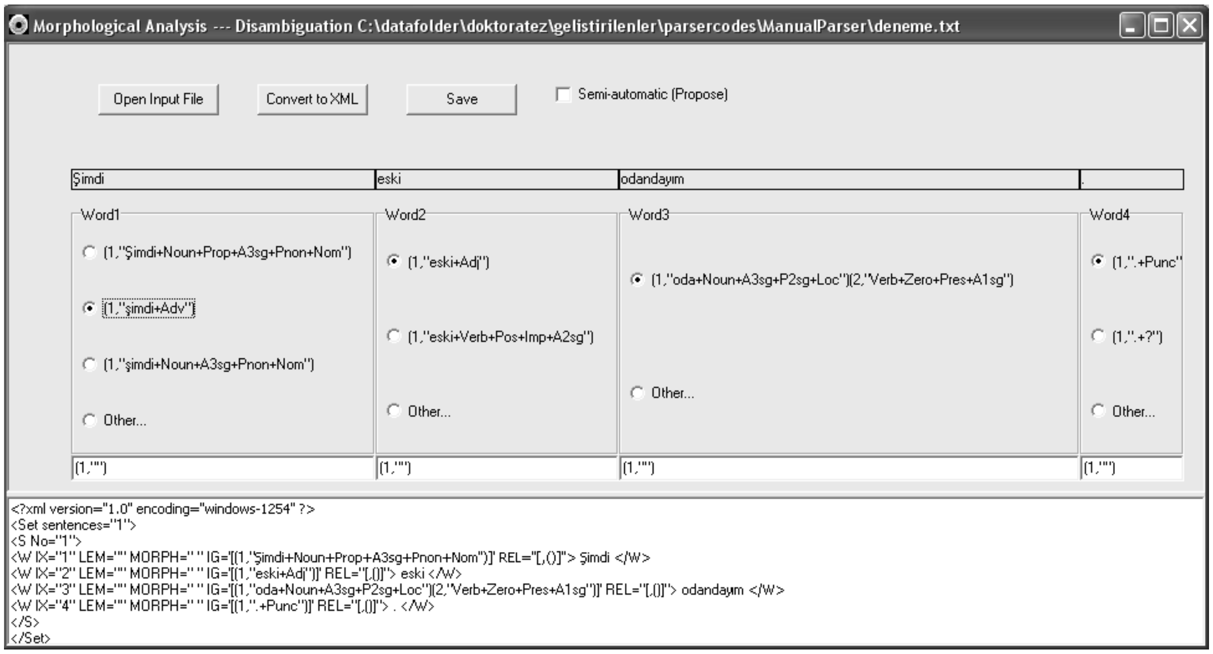

input
Stage 1
<automatic> Morphological Analysis
Plugin 1: Morphological Analyzer
Stage 2
<manual> Morphological Disambiguation
Plugin 2: POS tagger
Stage 3
<manual> Dependency Analysis
Plugin 3: Parser
output

Figure 1: Data Flow

ITU treebank annotation tool takes raw sentences as input and produces results in both the Turkish treebank original XML format (Atalay et al., 2003) and Conll treebank data format (Buchholz and


Morphological Analysis --- Disambiguation C:\datafolder\doktoratez\gelistirilenler\parsercodes\ManualParser\deneme.txt
Open Input File
Convert to XML
Save
Semi-automatic (Propose)
Word1
Word2
Word3
Word4

Şimdi | eski | odandayım | .

Word1: (1,"Şimdi+Noun+Prop+A3sg+Pnon+Nom") [✓ (1,"şimdi+Adv")] (1,"şimdi+Noun+A3sg+Pnon+Nom") Other... (1,"")

Word2: [✓ (1,"eski+Adj")] (1,"eski+Verb+Pos+Imp+A2sg") Other... (1,"")

Word3: [✓ (1,"oda+Noun+A3sg+P2sg+Loc")(2,"Verb+Zero+Pres+A1sg")] Other... (1,"")

Word4: [✓ (1,".+Punc"] (1,".+?") Other... (1,"")

```
<?xml version="1.0" encoding="windows-1254" ?>
<Set sentences="1">
<S No="1">
<W IX="1" LEM="" MORPH=" " IG='[(1,'Şimdi+Noun+Prop+A3sg+Pnon+Nom")]' REL="[,()]"> Şimdi </W>
<W IX="2" LEM="" MORPH=" " IG='[(1,"eski+Adj")]' REL="[,()]"> eski </W>
<W IX="3" LEM="" MORPH=" " IG='[(1,"oda+Noun+A3sg+P2sg+Loc")(2,"Verb+Zero+Pres+A1sg")]' REL="[,()]"> odandayım </W>
<W IX="4" LEM="" MORPH=" " IG='[(1,".+Punc")]' REL="[,()]"> . </W>
</S>
</Set>
```

Figure 2: Morphological Analysis and Disambiguation Screen

Marsi, 2006) which is now recognized by many of the state of the art dependency parsers.

The tool consists of three levels of annotation and can be used to produce results for each of them; these are morphological analysis, morphological disambiguation and syntax analysis stages. Each of these stages uses plugins in order to guide the human annotators (referred as *annotator*s in the remaining part). Figure 1 gives the data flow between the annotation stages and the plugins which will be explained in detail in the following sections.

3 Morphological Analysis

The most important characteristic of Turkish which distinguishes it from most of the well-studied languages is its very rich morphological structure. Turkish which is an agglutinative language has a very productive derivational and inflectional morphology. This rich structure of the language has been represented in the literature (Oflazer et al., 2003; Hakkani-Tür et al., 2002; Eryiğit and Oflazer, 2006) by splitting the words into inflectional groups (IGs) which are separated from each other by derivational boundaries. Each IG is then annotated with its own part-of-speech and inflectional features.

We are using the morphological analyzer of Oflazer (1994) which provides all the possible morphological analyses together with the IG structure. The output provided by the morphological analyzer for each word in the example sentence "*Şimdi eski odandayım.*" (I'm now in your old room.) can be seen from Figure 2 (the listed items under each word with radio buttons in front). We can see from the figure that the derived word "*odandayım*" (I'm in your room) is composed of two IGs:

$$\underbrace{\text{(1,"oda+Noun+A3sg+P2sg+Loc")}}_{IG_1}\ \underbrace{\text{(2,"Verb+Zero+Pres+A1sg")}}_{IG_2}$$

The first IG is the noun "*oda*" (room) which takes the meaning of "in your room" after taking the 3rd singular number-person agreement (+A3sg) , 2nd person possessive agreement (+P2sg) and locative case (+Loc) inflectional features. The second IG is the derived verb "being in your room" in present tense (+Pres), with 1st singular number-person agreement (+A1sg) inflectional features[1].

The morphological analysis stage is totally automatic except that the user can enter other analyses to the text boxes under each word if the correct one is not within the above listed items or the analyzer couldn't suggest any analysis. This latter case generally occurs for numerical values (e.g., numbers,

[1]+Zero means no additional suffix is used for the derivation.

dates) and unknown words. For numerical values, we use a preprocessor to produce the analysis, but for unknown words, the annotators are asked to enter the appropriate analysis.

4 Morphological Disambiguation

The second stage is the morphological disambiguation where the annotator is asked to choose one of the possible analyses for each word. The annotator may consult to an automatic analyzer by clicking the checkbox at the top of the screen in Figure 2. In this case we activate the part-of-speech tagger of Yüret and Türe (2006) which uses some rules automatically derived from a training corpus. The results of this tagger is reflected to the screen by selecting automatically the appropriate radio button for each word. After finishing the disambiguation, the annotator saves the results in XML format (shown at the bottom panel of Figure 2) and proceeds trough the syntax analysis.

5 Syntax Analysis

The syntactic annotation scheme used in the Turkish treebank is the dependency grammar representation. The aim of the dependency analysis is to find the binary relationships between dependent and head units. The dependency structure of Turkish has been mentioned in many studies (Oflazer et al., 2003; Oflazer, 2003; Eryiğit et al., 2006) and it is argued that for Turkish, it is not just enough to determine the relationships between words and one should also determine the relationships between inflectional groups. Figure 3 gives an example of this structure[2]. In this screen, the annotator first selects a dependent unit by selecting the check box under it and then a head unit and the appropriate dependency relation from the combo box appearing under the constructed dependency. In this figure, we see that the adjective "*eski*" (old) is connected to the first IG of the word "*odandayım*" since it is the word "*oda*" (room) which is modified by the adjective, not the derived verb form "*odandayım*" (I'm in your room). On the other hand, the adverb "*şimdi*" (now) is connected to the second IG of this word and modifies the verb "being in the room". The graphical interface is designed so that the annotator can easily determine the correct head word and its correct IG.

[2]The arrows in the figure indicates the dependencies emanating from the dependent unit towards the head unit.

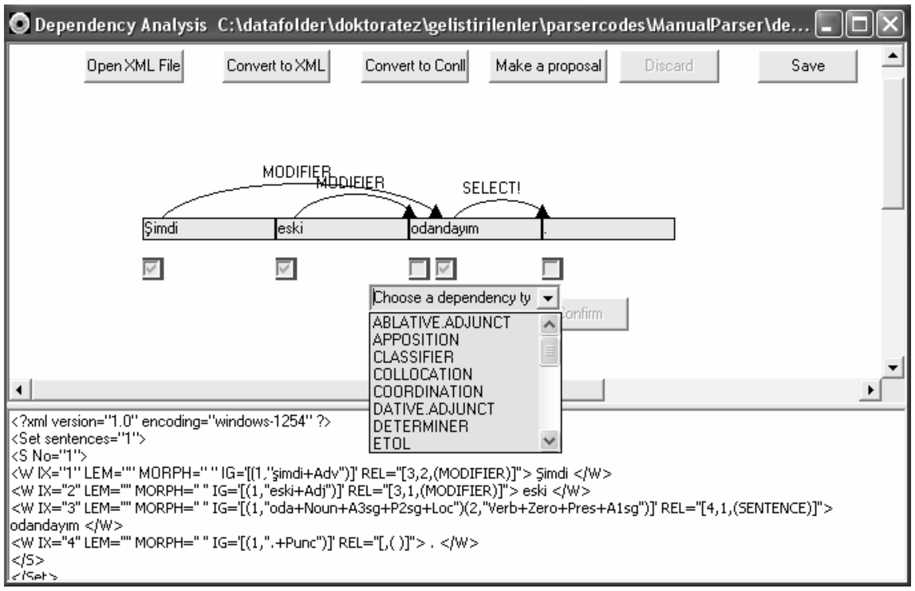
In each step of the syntactic annotation, the partially built dependency tree is shown to the annotators in order to reduce the number of mistakes caused by the inattentiveness of the annotators (such as the errors encountered in the original Turkish treebank; cycled dependencies, erroneous crossing dependencies, unconnected items, dependencies to nonexistent items). Extra cautions are taken with similar reasons in order to force the annotators to only make valid annotations:

- Only the check boxes under final IGs of the words become active when the annotator is about to select a dependent since the dependencies can only emanate from the last IGs of the dependents.
- The dependents may only be connected to the IGs of other words, thus the check boxes of the IGs within the dependent word become passive when selecting a head unit.

Similar to the morphological disambiguation stage, the annotator may want to consult to an automatic analyzer. We use the data-driven dependency parser of Nivre et al. (2006) as an external parsing guide which is shown to give the highest accuracy for Turkish and for many other languages. The output of the parser (pre-trained on the Turkish treebank) is reflected to the screen by automatically constructing the dependency tree. The annotator may then change the dependencies which he/she finds incorrect.

6 Conclusions and Future Work

ITU treebank annotation tool is a semi-automatic annotation tool tailored for the particular morphological structure of Turkish where we need to annotate units smaller than words. It has three annotation levels and uses pluggable analyzers in order to automate these levels. These are a rule-based morphological analyzer, and machine learning based part-of-speech tagger and dependency parser. The tool which aims to provide a user-friendly platform for the human annotators, also tries to minimize the number of errors due to the complexity of the annotation process of Turkish. The tool is designed and used only for Turkish in its current state, however it can be used for other languages with similar morphological structure (particularly other Turkic lan-



```
<?xml version="1.0" encoding="windows-1254" ?>
<Set sentences="1">
<S No="1">
<W IX="1" LEM="" MORPH=" " IG='[(1,"şimdi+Adv")]' REL="[3,2,(MODIFIER)]"> Şimdi </W>
<W IX="2" LEM="" MORPH=" " IG='[(1,"eski+Adj")]' REL="[3,1,(MODIFIER)]"> eski </W>
<W IX="3" LEM="" MORPH=" " IG='[(1,"oda+Noun+A3sg+P2sg+Loc")(2,"Verb+Zero+Pres+A1sg")]' REL="[4,1,(SENTENCE)]">
odandayım </W>
<W IX="4" LEM="" MORPH=" " IG='[(1,".+Punc")]' REL="[,( )]"> . </W>
</S>
</Set>
```

Figure 3: Dependency Analysis Screen

guages) by replacing the external analyzers. By using this tool, we observed significant acceleration both in correcting the existing treebank and developing new data sets. However education of new human annotators still remains as a difficult point and requires a lot of time. Hence in the future, we aim to develop online education tools which teach the annotators and tests their performance. We also aim to carry the platform to the web and supply an environment which can be reached from different places by volunteer researchers and collect the data in a single place.

Acknowledgment

The author wants to thank Kemal Oflazer for his valuable comments, Engin Taşkın and Cihat Eryiğit for their help during the development of the tool.

References

Nart B. Atalay, Kemal Oflazer, and Bilge Say. 2003. The annotation process in the Turkish treebank. In *Proc. of the 4th International Workshop on Linguistically Interpreteted Corpora*, Budapest.

Sabine Buchholz and Erwin Marsi. 2006. Conll-X shared task on multilingual dependency parsing. In *Proc. of the 10th CoNLL*, pages 149–164, New York, NY.

Gülşen Eryiğit and Kemal Oflazer. 2006. Statistical dependency parsing of Turkish. In *Proc. of the EACL*, pages 89–96, Trento.

Gülşen Eryiğit, Joakim Nivre, and Kemal Oflazer. 2006. The incremental use of morphological information and lexicalization in data-driven dependency parsing. In *Proc. of the ICCPOL*, pages 498–507, Singapore.

Dilek Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. 2002. Statistical morphological disambiguation for agglutinative languages. *Journal of Computers and Humanities*, 36(4):381–410.

Tuomo Kakkonen. 2006. Depann - an annotation tool for dependency treebanks. In *Proc. of the 11th ESSLLI Student Session*, pages 214–225, Malaga.

Joakim Nivre, Johan Hall, Jens Nilsson, Gülşen Eryiğit, and Stetoslav Marinov. 2006. Labeled pseudo-projective dependency parsing with support vector machines. In *Proc. of the CoNLL-X*, pages 221–225, New York, NY.

Kemal Oflazer, Bilge Say, Dilek Z. Hakkani-Tür, and Gökhan Tür. 2003. Building a Turkish treebank. In A. Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 261–277. Kluwer, Dordrecht/Boston/London.

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

Kemal Oflazer. 2003. Dependency parsing with an extended finite-state approach. *Computational Linguistics*, 29(4):515–544.

Deniz Yüret and Ferhan Türe. 2006. Learning morphological disambiguation rules for Turkish. In *Proc. of the HLT-NAACL*, pages 328–334, New York, NY.