Language Resources and Evaluation manuscript No.
(will be inserted by the editor)

Resources for Turkish Dependency Parsing

Introducing the BOUN Treebank and the BoAT Annotation Tool

Utku Türk · Furkan Atmaca · Şaziye Betül Özateş · Gözde Berk · Seyyit Talha Bedir · Abdullatif Köksal · Balkız Öztürk Başaran · Tunga Güngör · Arzucan Özgür

Received: date / Accepted: date

Abstract In this paper, we introduce the resources that we developed for Turkish dependency parsing, which include a novel manually annotated treebank (BOUN Treebank), along with the guidelines we adopted, and a new annotation tool (BoAT). The manual annotation process we employed was shaped and implemented by a team of four linguists and five Natural Language Processing (NLP) specialists. Decisions regarding the annotation of the BOUN Treebank were made in line with the Universal Dependencies (UD) framework as well as our recent efforts for unifying the Turkish UD treebanks through manual re-annotation. To the best of our knowledge, BOUN Treebank is the largest Turkish treebank. It contains a total of 9,761 sentences from various topics including biographical texts, national newspapers, instructional texts, popular culture articles, and essays. In addition, we report the parsing results of a state-of-the-art dependency parser obtained over the BOUN Treebank as well as two other treebanks in Turkish. Our results demonstrate that the unification of the Turkish annotation scheme and the introduction of a more comprehensive treebank lead to improved performance with regards to dependency parsing.

Keywords Turkish · Annotation · Treebanks · Dependency Syntax · Universal Dependencies · Resources

1 Introduction

The field of Natural Language Processing (NLP) has seen an influx of various treebanks following the introduction of the treebanks in Marcus et al. (1993), Leech and Garside (1991),

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant number 117E971 and as BIDEB 2211 graduate scholarship.

Utku Türk · Furkan Atmaca · Seyyit Talha Bedir · Balkız Öztürk Başaran
Deparment of Linguistics, Boğaziçi University
E-mail: utku.turk@boun.edu.tr, furkan.atmaca@boun.edu.tr, talha.bedir@boun.edu.tr, balkiz.ozturk@boun.edu.tr

Şaziye Betül Özateş · Gözde Berk · Abdullatif Köksal · Tunga Güngör · Arzucan Özgür
Department of Computer Engineering, Boğaziçi University
E-mail: saziye.bilgin@boun.edu.tr, gozde.berk@boun.edu.tr, abdullatif.koksal@boun.edu.tr, gungort@boun.edu.tr, arzucan.ozgur@boun.edu.tr

and Sampson (1995). These treebanks paved the way for today's ever-growing NLP framework, consisting of NLP applications, treebanks, and tools. Among the many languages with a growing treebank inventory, Turkish was one of the less fortunate languages. The latest version[1] of the Turkish IMST-UD Treebank is currently the 73rd largest treebank in terms of the number of annotated sentences in the Universal Dependencies (UD) project (Nivre et al. 2016). As of now, the UD project includes more than 150 treebanks and the largest of them, the UD German-HDT Treebank, consists of 190,000 sentences (Nivre et al. 2016). Due to its complex network of inflectional and derivational morphology, as well as its highly flexible word order, Turkish has posed an enormous challenge for NLP studies. One of the first attempts to create a structured treebank was initiated in the studies of Atalay et al. (2003) and Oflazer et al. (2003). Following these studies, many more Turkish treebanking efforts were introduced (Megyesi et al. 2010; Sulger et al. 2013; Sulubacak et al. 2016b, among others). However, most of these efforts contained a small volume of Turkish sentences, and some of them were re-introduced versions of already existing treebanks in another annotation scheme.

This paper aims to contribute to the limited NLP resources in Turkish by annotating a part of a brand new corpus that has not been approached with a syntactic perspective before, namely the Turkish National Corpus (henceforth TNC) (Aksan et al. 2012). TNC is an online corpus that contains 50 million words. The BOUN Treebank, which is introduced in this paper, includes 9,761 sentences extracted from five different text types in this corpus, i.e. essays, broadsheet national newspapers, instructional texts, popular culture articles, and biographical texts. These sentences have not been introduced within a treebank previously. We manually annotated the syntactic dependency relations of the sentences following the up-to-date UD annotation scheme.

Through a discussion of the annotation decisions made in the creation of the BOUN Treebank, we present our take on in the challenging issues in the annotation of Turkish data, including the copular clitic, embedded constructions, compounds, and lexical cases. Even though these linguistic phenomena are observed and studied extensively within Turkish linguistic studies, Turkish treebanking studies present an inconsistent picture in the annotation of such constructions.

In addition, we present a new annotation tool that integrates a tabular view, a hierarchical tree structure, and extensive morphological editing. We believe that other agglutinative languages that offer challenging morphological problems may benefit from this tool due to its ability to split and/or merge words and tokens in a sentence while rearranging the information regarding each word/token automatically, such as the word/token ID. This feature is crucial for the annotation process, since pre-processing of sentences may split the words and tokens erroneously.

Lastly, we report the results of an NLP task, namely dependency parsing, where we made parsing experiments on the newly introduced BOUN Treebank together with previous Turkish treebanks. The results show that increasing the size of the training set has a positive contribution to the parsing success for Turkish. We observe that using the UD annotation scheme more faithfully and in a unified manner within Turkish UD treebanks offers an increase in the UAS (Unlabeled Attachment Score) F1 and LAS (Labeled Attachment Score) F1 scores. We also report individual parsing scores for different text types within our new treebank.

This paper is organized as follows: In Section 2, we briefly explain the morphological and syntactic properties of Turkish. In Section 3, we present an extensive review of previous

[1] UD version 2.6. Available at http://hdl.handle.net/11234/1-3226

treebanking efforts in Turkish and locate them with regards to each other in terms of their use and their aim. In Section 4, we report the details of the BOUN Treebank and our annotation process including the morphological and syntactic decisions. We lay out our tool BoAT in Section 5 and, in Section 6, we report our experiments and their results. In Section 7, we present our conclusions and discuss the implications of our work.

2 Turkish

Turkish is a Turkic language spoken mainly in Asia Minor and Thracia with approximately 75 million native speakers. As an agglutinative language, Turkish makes excessive use of morphological concatenation. According to Bickel and Nichols (2013), a Turkish verb may have up to 8-9 inflectional categories per word, such as number, tense, or person marking, which is above the average 3-4 inflectional categories per word. The number of morphological categories increases even more when considering derivational processes. Kapan (2019) states that Turkish words may host up to 6 different derivational affixes at the same time. The complexity of morphological analysis, however, is not limited to the sheer numbers of inflectional and derivational affixes. In addition to such affixes, allomorphies, vowel harmony processes, elisions, and insertions create an arduous task for researchers in Turkish NLP. Table 1 lists the possible morphological analyses of the surface word *alın*. The table shows that despite the shortness of the word, the morphological analysis is toilsome; and even such a short item may be parsed to have different possible roots.

Table 1 Possible morphological analyses of the word *alın* from Sak et al. (2008). The symbol ‘&’ indicates derivational morphemes (originally ‘-’, changed for clarity here), and ‘+’ indicates inflectional morphemes. The strings between these symbols and the square-bracketed feature represent the phonology of the suffix. Upper case within a suffix means that the sound is phonologically conditioned. ‘H’ stands for the archiphonemic high vowel. ‘N’ stands for the allomorphy between the alveolar nasal and the lack of it. ‘Y’ represents the allomorphy between the palatal glide and the lack of it.

Root	Category of the root	Features	Gloss	Transliteration
alın	[Noun]	+[A3sg]+[Pnon]+[Nom]	forehead	’forehead’
al	[Noun]	+[A3sg]+Hn[P2sg]+[Nom]	red-POSS	’your red color’
al	[Adj]	&[Noun]+[A3sg]+Hn[P2sg]+[Nom]	red-POSS	’your red color’
al	[Noun]	+[A3sg]+[Pnon]+NHn[Gen]	red-GEN	’belonging to the color red’
al	[Adj]	&[Noun]+[A3sg]+[Pnon]+NHn[Gen]	red-GEN	’belonging to the color red’
alın	[Verb]	+[Pos]+[Imp]+[A2sg]	offend-2SG.IMP	’Get offended!’
al	[Verb]	+[Pos]+[Imp]+YHn[A2pl]	take-2SG.HNR-IMP	’(Please) take it!’
al	[Verb]	&Hn[Verb+Pass]+[Pos]+[Imp]+[A2sg]	take-PASS[2SG]	’Get taken!’

With respect to syntactic properties, Turkish has a relatively free word order, which is constrained by discourse elements and information structure (Taylan 1986; Hoffman 1995; Kural 1997; İşsever 2003; Kornfilt 2005; Öztürk 2008, 2013; Özsoy 2019). For example, new information introduced are placed post-verbally. Even though SOV is the base word order, other permutations are highly utilized, as exemplified in Example 1.[2] The percentages were determined by Slobin and Bever (1982) from 500 utterances of spontaneous speech. We also report word order percentages acquired from the BOUN Treebank in Table 13 and Table 14 in Appendix B. These permutations are stemmed from processes including topicalization, focusing, and backgrounding. Contributing new or old information may also affect the place of a constituent, that is, new information may be placed closer to the verb and in always pre-verbal position, whereas the old information may surface both in preverbal and post-verbal positions. Another aspect that affects the word order is definiteness and specificity Indefinite subjects and objects can typically surface in the immediately preverbal position.

(1) a. *Fatma Ahmet'i gör-dü.* (SOV 48%)
Fatma Ahmet-ACC see-PST
'Fatma saw Ahmet.'
b. Ahmet'i Fatma gördü. (OSV 8%)
c. Fatma gördü Ahmet'i. (SVO 25%)
d. Ahmet'i gördü Fatma. (OVS 13%)
e. Gördü Fatma Ahmet'i. (VSO 6%)
f. Gördü Ahmet'i Fatma. (VOS <1%) (adapted from Hoffman 1995)

As for the case system, every element in a sentence needs to host a case according to its syntactic role, semantic contribution, or the lexical selection of the phrasal head (Taylan 2015). These groupings, however, are not clear cut and there is not always a one-to-one correspondence between cases and their roles.

Moreover, Turkish is a pro-drop language in which the subject can be elided when it is retrievable from the given discourse (Kornfilt 1984; Özsoy 1988). Overt subjects are used only to convey certain discourse and/or pragmatic effects, such as a change in context or focus. However, the subject is also retrievable from the agreement marker on the verb. In addition to these properties, Turkish is also a null object language, even though the language does not have an overt agreement marker available for this process (Öztürk 2006). If the object of a sentence is retrievable from the given discourse, speakers may omit the object without any overt marking on the verb. The final issue with Turkish syntax lies in the fact that it frequently makes use of nominalization processes for embedded clauses (Göksel and Kerslake 2005). When followed by certain suffixes, the whole embedded structure can behave as any part of the sentence. This sentence embedding strategy complicates the annotation process since the final form of the construction is of a different category derived from a verb. However, these constructions encode complex predication and may act as a subject, object, adjective, adverb or even predicate on their own.

3 Previous Turkish Treebank Initiatives

Following the studies on treebanks for languages such as English, Chinese, Arabic, and many more (Leech and Garside 1991; Marcus et al. 1993; Sampson 1995; Maamouri et al.

[2] Abbreviations used in the paper are as follows: 1 = first person, 2 = second person, 3 = third person, ABL = ablative, ACC = accusative, AOR = aorist, CAUS = causative, COM = comitative, COND = conditional, COP = copula, CVB = converb, DAT = dative, EMPH = emphasis, FUT = future, GEN = genitive, HNR = honorific, IMP = imperative, LOC = locative, NEG = negative, NMLZ = nominalizer, PASS = passive, PL = plural, POSS = possessive, PROG = progressive, PST = past, Q = question particle, SG = singular

2004; Xue et al. 2005), the initial groundwork for Turkish treebanks was laid in Atalay et al. (2003) and Oflazer et al. (2003). The first of its kind, the Metu-Sabancı Treebank (MST) consists of 5,635 sentences, a subset of the METU corpus that includes 16 different text types such as newspaper articles and novels (Say et al. 2002). Oflazer et al. (2003) encoded both morphological complexities and syntactic relations. Due to the productive use of derivational suffixes, they explicitly spelled out every inflection and derivation within a word. As for the syntactic representation, Atalay et al. (2003) used a dependency grammar in order to bypass the problem of constituency in Turkish, which arises from the relatively free word order of the language.

Branching off the work of Atalay et al. (2003) and Oflazer et al. (2003), a small treebank with the name of ITU Validation set for MST was introduced. It contains 300 sentences from 3 different genres. The treebank was introduced as a test set for MST in the CoNLL-XI shared task (Eryiğit and Pamay 2007). The treebank was annotated by two annotators using a cross-checking process. Following this work, MST was re-annotated by Sulubacak et al. (2016a) from ground up with revisions made in syntactic relations and morphological parsing. The latest version was renamed as the ITU-METU-Sabancı Treebank (IMST). Due to certain limitations, Sulubacak et al. (2016a) employed only one linguist and several NLP specialists. The annotation process was arranged in such a way that there was no cross-checking between the works of the annotators. Moreover, inter-annotator agreement scores, details regarding the decision process among annotators, and the adjudication process have not been reported. Nevertheless, this re-annotation solved many issues regarding MST by proposing a new annotation scheme. Even though problems such as semantic incoherence in the usage of annotation tags and ambiguous annotation were resolved to a great extent, the non-communicative nature of the annotation process led to a handful of inconsistencies.

The inconsistencies in IMST were also carried over to IMST-UD, which utilizes automatic conversions of the tags from IMST to the UD framework (Sulubacak et al. 2016b). Mappings of syntactic and morphological representations were also included. Following such changes, IMST-UD was made more explanatory and clear thanks to the systematically added additional dependencies. While IMST had 16 dependency relations, 47 morphological features, and 11 parts of speech types, IMST-UD upped these numbers to 29, 67, and 14, respectively. However, the erroneous dependency tagging resulting from morphophonological syncretisms lingered long after the publication of the treebank. Moreover, no post-editing effor has been reported. Even though there have been four updates since the first release of the IMST-UD treebank, there are still mistakes that can easily be corrected through a post-editing process, such as the punctuation marks tagged as roots, reversed head-dependent relations, and typos in the names of syntactic relations.

Apart from the treebanks originating from MST, many other treebanks have emerged. Some of these treebanks can be grouped under the class of *parallel treebanks*. The first of these parallel treebanks is the Swedish-Turkish parallel treebank (STPT). Megyesi et al. (2008) published their parallel treebank containing 145,000 tokens in Turkish and 160,000 tokens in Swedish. Following this work, Megyesi et al. (2010) published the Swedish-Turkish-English parallel treebank (STEPT). This treebank includes 300,000 tokens in Swedish, 160,000 tokens in Turkish, and 150,000 tokens in English. Both of these treebanks utilized the same morphological and syntactical parsing tools. For Swedish morphology, the Trigrams 'n' Tags tagger (Brants 2000) trained on Swedish (Megyesi 2002), was used. On the other hand, Turkish data were first analyzed using the parser in Oflazer (1994), and its accuracy was enhanced through the morphological disambiguator proposed in Yuret and Türe (2006). Both of them were annotated using the MaltParser (Nivre et al. 2007) and were trained with the Swedish treebank Talbanken05 (Nivre et al. 2006) and MST (Oflazer et al. 2003), respectively.

Another parallel treebank introduced for Turkish is the Turkish PUD Treebank, which adopts the UD framework. The Turkish PUD Treebank was published as part of a collaborative effort, the CoNLL 2017 Shared Task on Multilingual Parsing from Raw Text to Universal Dependencies (Zeman et al. 2017). Sentences for this collaborative treebank were drawn from newspapers and Wikipedia. The same 1,000 sentences were translated into more than 40 languages and manually annotated in line with the universal annotation guidelines of Google. After the annotation, the Turkish PUD Treebank was automatically converted to the UD style.

Moreover, there are three treebanks that consist of informal texts. One such treebank was introduced by Pamay et al. (2015) under the name of ITU Web Treebank (IWT). In IWT, non-canonical data was included such as the usage of punctuations as in emoticons, abbreviated writing such as *kib* that stands for *kendine iyi bak* ('take care of yourself'), and non-standard writing conventions as in *saol* instead of *sağol* ('thanks'). Later on, the UD version of IWT was also introduced (Sulubacak and Eryiğit 2018a). Another web treebank has recently been presented by Kayadelen et al. (2020), which is larger than the previous Turkish treebanks in terms of word count, but still smaller than the BOUN Treebank that we introduce in this paper. Kayadelen et al. (2020) used a set of dependency labels similar to the UD framework. However, they diverge from the UD framework in certain issues such as postpositions, indirect objects, and oblique arguments. The Turkish-German Code-Switching Treebank (Çetinoğlu and Çöltekin 2016) is also made out of informal text from twitter. This treebank includes 1,029 bilingual Turkish-German tweets that have been annotated with respect to the language in use and the UD part of speech tags.

There is also one grammar book-based treebank introduced in Çöltekin (2015). The Grammar Book treebank (GB) is the first UD attempt in Turkish treebanking. In this treebank, data were collected from a reference grammar book for Turkish written by Göksel and Kerslake (2005). It includes 2,803 items that are either sentences or sentence fragments from the grammar book. It utilized TRMorph (Çöltekin 2010) for morphological analyses and the proper morphological annotations were manually selected amongst the suggestions proposed by TRMorph. The sentences were manually annotated in the native UD-style.

In addition to these treebank initiatives, we recently started our unifying efforts in the syntactic annotation scheme in Turkish treebanking. We manually corrected the syntactic annotations in the Turkish PUD and IMST-UD treebanks (Türk et al. 2019a,b). In these works, we selected the treebanks that were not annotated natively in the UD style and unified the annotation scheme. This process improved the UAS score for the IMST-UD Treebank from 72.49 to 75.49 and caused only a 0.9 point decrease in the LAS score (from 66.43 to 65.53) in our experiments with the Standford's neural dependency parser (Dozat et al. 2017), although the number of unique dependency tags increased from 31 to 40 with the newly included dependency types (Türk et al. 2019a). On the other hand, there was a decrease in the parsing accuracy for the re-annotated version of the PUD Treebank in terms of the attachment scores. While the parser achieved an UAS score of 79.52 and a LAS score of 73.81 on the previous version of the PUD Treebank, its attachment scores for the re-annotated version are 78.70 UAS and 70.01 LAS (Türk et al. 2019b). We want to note that, we used 5-fold cross validation for the evaluation on the PUD Treebank due its small size. In each fold, the parser had only 600 sentences for training and 200 sentences were used as development set. The evaluation was done on the remaining 200 sentences. The small size of the PUD Treebank, which was originally used only for evaluation purposes (not for training) in the CoNLL 2017 Shared Task (Zeman et al. 2017), renders the results less reliable. Following these studies, with the annotation scheme we unified, we manually annotated the BOUN Treebank, which we present in this paper.

4 The BOUN Treebank[3]

In this paper, we introduce a treebank that consists of 9,761 sentences which form a subset of the Turkish National Corpus (TNC) (Aksan et al. 2012). TNC includes 50 million words from various text types, and encompasses sentences from a 20 year period between 1990 and 2009. The principles of the British National Corpus were followed in terms of the selection of the domains. Table 2 shows the percentages of different domains and media used in TNC.

Table 2 Composition of the written component of TNC, adapted from Aksan et al. (2012).

Domain	%	Medium	%
Imaginative	19	Books	58
Social Science	16	Periodicals	32
Art	7	Miscellaneous published	5
Commence/Finance	8	Miscellaneous unpublished	3
Belief and Thought	4	Written-to-be-spoken	2
World Affairs	20		
Applied Science	8		
Nature Science	4		
Leisure	14		

In our treebank, we included the following text types: essays, broadsheet national newspapers, instructional texts, popular culture articles and biographical texts. Approximately 2,000 sentences were randomly selected from each of these registers. All of the selected sentences were written items and were not from the spoken medium. Our motivation for using these registers was to cover as many domains as possible using as few registers as possible, while not compromising a variation in length, formality, and literary quality. TNC consists of 39 different registers, reported in Table 15 in Appendix C.[4] Sampling our sentences from all of the registers available in TNC would result in a treebank that is inconsistent due to the small sample size of the existing registers. The basic statistics for the BOUN Treebank and its different sections are provided in Table 3.

Before the manual annotation of the BOUN Treebank, the sentences were first automatically annotated using a complete parsing pipeline tool that parses raw texts to UD dependencies in CoNLL-U format with POS and morphological tagging information (Kanerva et al. 2018). The manual syntactic annotation of sentences were then performed on this automatically generated CoNLL-U versions of the corpus sentences. In the manual annotation process, we followed the UD syntactic relation tags. First, we reviewed the dependency relations in use within the UD framework. Upon reviewing the definitions, we created a list of unique sentences that we believe are representative of the UD dependency relations in Turkish. Later on, we compared our sentences with the examples from already existing Turkish UD treebanks. If found problematic, the definition of a dependency relation has been discussed with all the linguists within the team.

[3] Our treebank is available online in `https://github.com/boun-tabi/UD_Turkish-BOUN`

[4] This table is retrieved from `https://www.tnc.org.tr/about-the-corpus/object/` in September 15 2020.

Table 3 Word statistics of the different sections of the BOUN Treebank. The difference between the numbers of tokens and words is due to multi-word expressions being represented with a single token, but with multiple words.

Treebank	Num. of sentences	Num.of tokens	Num. of word forms
Essays	1,953	27,007	27,557
Broadsheet National Newspapers	1,898	29,307	29,386
Instructional Texts	1,976	20,442	20,625
Popular Culture Articles	1,962	21,067	21,263
Biographical Texts	1,972	23,391	23,553
Total	**9,761**	**121,214**	**122,384**

After settling on the definitions of dependency relations, two Turkish native speaker linguists manually annotated the BOUN Treebank using our annotation tool that is presented in Section 5. Following the annotation process, two other linguists who were not assigned parts of the BOUN Treebank cross-checked the syntactic annotations of the two linguists. When a problematic sentence or an inconsistency was encountered, discussions with regards to the sentence and related sentences were held among the team members. After a decision was made, the necessary changes were applied uniformly. In addition to the cross-checking process, the annotators also performed an additional annotation independently for the same set of 1,000 randomly selected sentences. Table 4 shows the Cohen's Kappa measures of inter-annotator agreement for finding the correct heads (κ_{Head}) and the correct dependency label of the syntactic relations (κ_{Label}). This multiple annotation part was the first part both annotators completed. The disagreements were discussed and resolved with the entire team of linguists and NLP specialists.

Table 4 The Cohen's Kappa measures of inter-annotator agreement with regards to head-dependent relation and dependency tags.

Annotator Pair	κ_{Head}	κ_{Label}
1-2	0.82	0.83

4.1 Levels of Annotation

4.1.1 Morphology

Turkish makes use of affixation much more frequently than any other word-formation process. Even though it adds an immense complexity to its word level representation, patterns within the Turkish word-formation process allowed previous research to formulate morphological disambiguators that dissect word-level dependencies. One such work was introduced by Sak et al. (2011). Their morphological parser is able to run independently of any other external system and is capable of providing the correct morphological analysis with 98% accuracy using contextual cues, such as the two previous tags.

In the morphological annotation of the BOUN Treebank, we decided to use the morphological analyzer and the disambiguator of Sak et al. (2011). For this purpose, the tokenized sentences were first given to the morphological parser of Sak et al. (2011). The output of the parser was converted to the corresponding UD features automatically. In rare cases where the morphological parser did not return a morphological analysis for a token, the morphological features column from Turku pipeline (Kanerva et al. 2018) for this token was used. The same operation was done for the lemmas of the tokens as well.

Our preference for the morphological tagger of Sak et al. (2011) instead of the morphological tagger of the Turku parsing pipeline (Kanerva et al. 2018) that we used for the automatic processing of the treebank in the first step is due to their comparison in terms of the token-based accuracy, and the feature-based recall, precision, and f-measure metrics. After randomly selecting 50 words from every text type in the BOUN Treebank (a total of 250 words for the five text types), we encoded the errors made by the morphological parsers. The results are shown in Table 5. *Token Accuracy* column represents the token-based accuracy, namely the percentage of words for which correct morphological analyses are produced. *Recall* column represents the ratio of the number of correct morphological features to the number of morphological features in the gold standard. *Precision* column encodes the ratio of the number of correct morphological features to the total number of morphological features predicted by the morphological parser. The *F1-measure* column is the harmonic mean of precision and recall. Our scores align with the scores reported in the original study of Sak et al. (2011), even though their training set and our set here consist of different text types. While they only used newspaper corpora in the training set, we replicated their results on different text types including essays, instructional texts, biographical texts, and popular culture articles.

Table 5 The performance of Sak et al. (2011)'s and Turku pipeline's (Kanerva et al. 2018) morphological taggers for BOUN Treebank.

Morphological Tagger	Token Accuracy	Recall	Precision	F1-measure
Sak et al. (2011)	0.91	0.94	0.95	0.94
Turku pipeline	0.82	0.89	0.83	0.86

The morphological parser of Sak et al. (2011) does not provide morphological tags in UD format. So, we automatically converted its output to the UD format. In this process, we maximally used the morphological features from the UD framework. When there is no clear-cut mapping between the features that we acquired from the morphological parser of Sak et al. (2011) and features proposed in the UD framework, we used the features previously suggested in the works of Çöltekin (2016), Tyers et al. (2017b), and Sulubacak and Eryiğit (2018b). These features were already stated in the UD guidelines. Table 12 in Appendix A shows the automatic conversion from the results of Sak et al. (2011)'s morphological disambiguator. Due to varying linguistic concerns, the depth of morphological representation in Sak et al. (2011) and that in the UD framework do not align perfectly. When necessary, we used the morphological cues provided by the morphological parser to decide on UPOS and lemma.

So, in our treebank, in addition to strings of words, we encoded the lexical and grammatical properties of the words as sets of features and values for these features. We also encoded the lemma of every word separately, following the UD framework. Table 6 shows an example sentence encoded with the CoNLL-U format.

Table 6 An example sentence from our treebank encoded in CoNLL-U format.

<table>
<tr><td colspan="10"># sent_id = ins_167</td></tr>
<tr><td colspan="10"># text = Sözü uzatıp seni merakta bıraktım galiba.</td></tr>
<tr><td colspan="10"># trans = Probably, I beat around the bush and kept you in suspense.</td></tr>
<tr><th>ID</th><th>FORM</th><th>LEMMA</th><th>UPOS</th><th>XPOS</th><th>FEATS</th><th>HEAD</th><th>DEPREL</th><th>DEPS</th><th>MISC</th></tr>
<tr><td>1</td><td>Sözü</td><td>söz</td><td>NOUN</td><td>Noun</td><td>Case=Acc|Number=Sing|Person=3</td><td>2</td><td>obj</td><td>_</td><td>_</td></tr>
<tr><td>2</td><td>uzatıp</td><td>uza</td><td>VERB</td><td>Verb</td><td>Polarity=Pos|Voice=Cau</td><td>5</td><td>advcl</td><td>_</td><td>_</td></tr>
<tr><td>3</td><td>seni</td><td>sen</td><td>PRON</td><td>Pers</td><td>Case=Acc|Number=Sing|Person=2</td><td>5</td><td>obj</td><td>_</td><td>_</td></tr>
<tr><td>4</td><td>merakta</td><td>merak</td><td>NOUN</td><td>Noun</td><td>Case=Loc|Number=Sing|Person=3</td><td>5</td><td>obl</td><td>_</td><td>_</td></tr>
<tr><td>5</td><td>bıraktım</td><td>bırak</td><td>VERB</td><td>Verb</td><td>Aspect=Perf|Evident=Fh|Number=Sing|
Person=1|Polarity=Pos|Tense=Past</td><td>0</td><td>root</td><td>_</td><td>_</td></tr>
<tr><td>6</td><td>galiba</td><td>galiba</td><td>ADV</td><td>Adverb</td><td>_</td><td>5</td><td>advmod</td><td>_</td><td>SpaceAfter=No</td></tr>
<tr><td>7</td><td>.</td><td>.</td><td>PUNCT</td><td>Punc</td><td>_</td><td>5</td><td>punct</td><td>_</td><td>SpacesAfter=\n</td></tr>
</table>

4.1.2 Syntax

In the BOUN Treebank, we decided to represent the relations amongst the parts of the sentences within a dependency framework. This decision has two main reasons. The main and the historical reason is the fact that the growth of Turkish treebanks has been mainly within the frameworks where the syntactic relations have been represented with dependencies (Oflazer 1994; Çetinoğlu 2009). The other reason is the fact that Turkish allows for phrases to be scrambled to pre-subject, post-verbal, and any clause-internal positions with specific constraints, which makes building constituency grammars quite difficult (Taylan 1984; Kural 1992; Aygen 2003; İşsever 2007). With these in mind, we wanted to stick with the conventional dependency framework and use the recently rising UD framework.[5] One of the main advantages of the UD framework is that it creates directly comparable sets of treebanks with regards to their syntactic representation due to its very nature.

By following the UD framework, we implicitly encode two different syntactic information for each dependent: the category of the dependent and the function of this dependent with regards to its syntactic head. This is due to the grouping of dependency relations introduced by the UD framework. The selection of the syntactic dependency relation for each dependent is mainly based on the functional category of the dependent in relation to the head and the structural category of the head. In terms of the functional category of the dependent, UD framework differentiates core arguments of clauses, non-core arguments of clauses, and dependents of nominal heads. As for the category of the dependent, the UD framework makes use of a taxonomy that distinguishes between function words, modifier words, nominals, and clausal elements. In addition to this classification there are some other groupings which may be listed as: coordination, multiword expressions, loose syntactic relation, sentential, and extra-sentential.[6] Table 7 shows the dependency relations that we employed in this treebank with their counts and percentages.

Every dependency forms a relation between two segments within the sentence, building up to a non-binary and hierarchical representation of the sentence, that is every node can have more than 2 children nodes and every node is accessible from the root node. This representation is exemplified in Example 2 [7] using the sentence in Table 6.

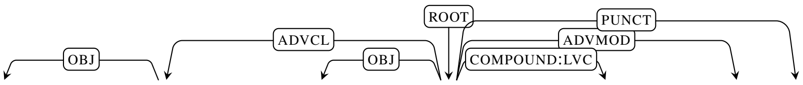
[5] For more information on the UD framework see `https://universaldependencies.org/u/dep/index.html`. For our annotation guidelines please see `https://github.com/boun-tabi/UD_docs/blob/main/_tr/dep/Turkish_deprel_guidelines.pdf`.

[6] For the complete table of syntactic relations, please check `https://universaldependencies.org/u/dep/index.html`

[7] We use Leipzig glossing conventions (Comrie et al. 2008) in our examples. A dash "-" separates affixes, equal "=" separates a clitic, and "~" is used for non-separable but identifiable changes like base modification.

Table 7 The dependeny relation set of the BOUN Treebank.

Relation Type	Count	Percentage	Relation Type	Count	Percentage
`acl`	3,500	2.86%	`det`	4,936	4.03%
`advcl`	2,589	2.12%	`discourse`	377	0.31%
`advcl:cond`	268	0.22%	`dislocated`	28	0.02%
`advmod`	5,277	4.31%	`fixed`	12	0.01%
`advmod:emph`	1,721	1.41%	`flat`	2,033	1.66%
`amod`	7,864	6.43%	`goeswith`	4	0.002%
`appos`	506	0.41%	`iobj`	165	0.13%
`aux`	39	0.03%	`list`	40	0.03%
`aux:q`	269	0.22%	`mark`	117	0.10%
`case`	3,303	2.7%	`nmod`	1,386	1.13%
`cc`	2,799	2.29%	`nmod:poss`	10,392	8.49%
`cc:preconj`	134	0.11%	`nsubj`	8,498	6.94%
`ccomp`	1,510	1.23%	`nummod`	1,567	1.28%
`clf`	122	0.1%	`obj`	7,379	6.03%
`compound`	2,382	1.95%	`obl`	12,009	9.81%
`compound:lvc`	1,218	1.0%	`orphan`	83	0.07%
`compound:redup`	456	0.37%	`parataxis`	208	0.17%
`conj`	7,248	5.92%	`punct`	20,116	16.44%
`cop`	1,291	1.05%	`root`	9,761	7.97%
`csubj`	545	0.45%	`vocative`	87	0.07%
`dep`	9	0.01%	`xcomp`	125	0.01%



(2) Söz-ü uza-t-ıp sen-i merak-ta bırak-tı-m galiba .
word-ACC strech-CAUS-NMLZ you-ACC curiosity-LOC leave-PST-1SG probably .
'Probably, I beat around the bush and kept you in suspense.'

4.2 Corrections in the Annotation Process

In the annotation process of the BOUN Treebank, we stayed faithful to the UD main tag set and the previous conventions of Turkish annotation schemes for the most part. However, there were some instances where we diverge from these conventions and offer a new analysis. They mostly stem from the poor maintenance of the previous Turkish UD treebanks. In this section, we provide the our linguistic decision process for these instances. Our decisions are in the same spirit of unifying the annotation scheme within Turkish UD treebanks, which was done in our previous works (Türk et al. 2019a,b). Our main concern is to reflect linguistic adequacy in the BOUN Treebank following Manning's Law (Nivre et al. 2017). We paid great attention to follow the previous discussion within the UD framework, such as the discussion on the copular clitic and the objecthood-case marking relation. In the following sections, we first touch upon the issues where we believe the previous conventions were erroneous or there were mistakes of poor maintenance in Turkish UD treebanks. These issues include the annotation of the embedded sentences, the treatment of copular verb, and the analysis of compounds. In the second part, we discuss the issue of objecthood and the case marking relation in Turkish, where we adopt a simpler analysis that has been used in other dependency grammars instead of recently discussed UD alternatives.

4.2.1 Annotation of Embedded Clauses

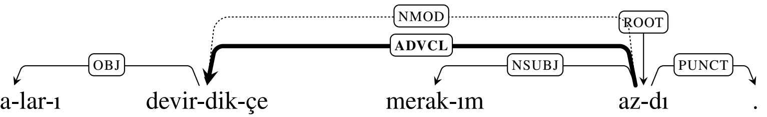
The first issue for which we have laid out a clear annotation process is the annotation of embedded clauses. In the previous treebanks, the annotation of embedded clauses did not reflect the inner hierarchy that a clause by definition possesses. This is mostly due to the morphological aspect of the common embedding strategy in Turkish: nominalization. Due to nominalization, embedded clauses in Turkish can be regarded as nominals since they behave like nominals: They may be marked with an accusative case, can be substituted with another nominal, and marked for person like in genitive-possessive cases as shown in Example 3. The embedded clause in the given sentence is shown with square brackets. The input within the brackets can be replaced with a simple noun, like *otobüs* (bus), or a complex noun phrase *senin otobüsün* (your bus) as in Example 4.

(3) [*Sen-in* *otobüs-ü* *sür-düğ-ün*]*-ü* *gör-dü-m.*
[you-GEN bus-ACC drive-NMLZ-POSS]-ACC see-PST-1SG
'I saw that you drove the bus.'

(4) *Sen-in* *otobüs-ün-ü* *gör-dü-m.*
you-GEN bus-POSS-ACC see-PST-1SG
'I saw your bus.'

Previous treebanks in the UD framework used dependency relations such as `obj`, `nsubj`, `amod`, or `advmod` to mark the relation of the embedding with the matrix verb. These were used instead of relations like `ccomp`, `csubj`, `acl`, or `advcl` due to the surface form and behaviour of the embedded clauses. In our annotation process, we wanted to emphasize the clausal nature of these embedded clauses by focusing on their internal structure and highlight the existence of a temporal domain within. For instance, Example 3 would be ungrammatical if we had the time adverb *tomorrow* within the embedded clause. This ungrammatically is due to the tense information introduced by the nominalizer '-düğ' in the example. If there would be an adverb like tomorrow in an embedded clause marked with '-düğ', the previous annotation scheme would not be able to detect the ungrammaticality. In our annotation scheme, the ungrammaticality is borne out naturally since the annotation respects the clause-hood of the embedding.

This argumentation applies to converbs as well. Converbs are verbal elements of a non-finite adverbial clause (Göksel and Kerslake 2005). They may act as adverbial adjuncts or as discourse connectives. In the previous annotation processes in Turkish, they were annotated as `nmod`. The reason behind this is again the fact that they behave like nominals; they may be marked with inflectional and derivational suffixes that normally nouns bear. Considering their clausal properties, such as their temporal domain, ability to host a subject, an object, and a tense/aspect/modality information, we annotated them as `advcl` like in Example 5. [8]

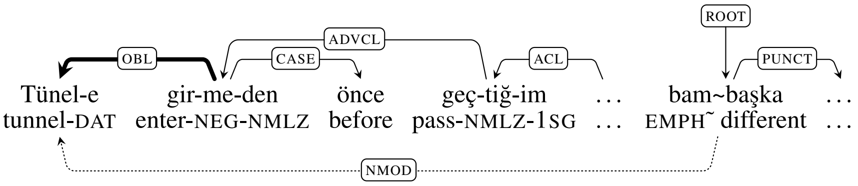


(5) Bira-lar-ı devir-dik-çe merak-ım az-dı .
beer-PL-ACC topple-NMLZ-CVB curiosity-1SG.POSS get.wild-PSTW .
"As I finish my beers, my curiosity peaked."

[8] Throughout the paper, changes in the annotation convention introduced by us are shown with bold arcs, whereas the dashed arcs suggest previous annotations. Default arcs represent unaltered dependencies. Every annotated tree that contains a bold arc in this paper is taken from previous Turkish Treebanks, that is either the IMST-UD Treebank or the Turkish PUD Treebank.

In addition to the annotation of the whole embedded clause, dependents within the embedded clause were erroneously annotated in the previous Turkish annotation schemes. For example, an oblique of an embedded verb used to be attached to the root since the embedded verb is seen as a nominal, and not as a verb as in Example 6.

(6)

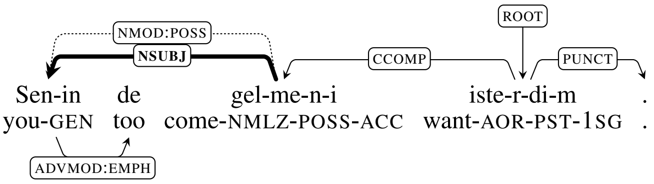


Tünel-e gir-me-den önce geç-tiğ-im ... bam~başka ...
tunnel-DAT enter-NEG-NMLZ before pass-NMLZ-1SG ... EMPH~ different ...

'The scenery that I passed before I entered the tunnel was completely different from here.'

Likewise, the genitive subjects of embedded clauses were wrongly marked as a possessive nominal modifier, whereas they are one of the obligatory elements of the embedded structures. This wrong annotation in the previous treebanks is due to the fact that Turkish makes use of genitive-possessive structure for marking the agreement in an embedded clause as in Example 7 (Göksel and Kerslake 2005). Despite the morphology, the word *senin* here serves as subject. Example 8 shows the causativized version of the embedded verb in sentence Example 7. When we causativize the subject of an intransitive verb, we expect the subject to be marked with an accusative case and act as an direct object. As seen below, the work *sen* reflects the morphological reflex stemming from a syntactic voice change. Thus, it cannot be a modifier and it has to be an argument.

(7)



Sen-in de gel-me-n-i iste-r-di-m .
you-GEN too come-NMLZ-POSS-ACC want-AOR-PST-1SG .

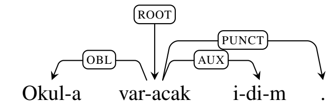
'I would have wanted you to come, as well.'

(8) *O-nun sen-i de getir-me-si-ni iste-r-di-m.*
she-GEN you-ACC too come.CAUS-NMLZ-POSS-ACC want-AOR-PST-1SG.
'I would have wanted her to brıng you, as well.'

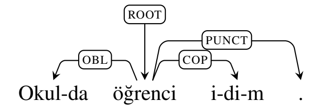
Due to the reasons explained above, in the annotation of embedded clauses we used dependency relations that emphasize the clausal nature of nominalized verbs, i.e., `csubj`, `ccomp`, `advcl`, instead of dependency relations that emphasize the final product of local derivations, i.e., `nsubj`, `obj`, `advmod`, respectively.

4.2.2 Copular and Auxiliary Clitic

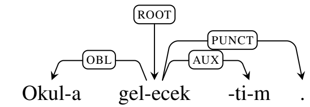
Due to its agglutinating nature, the line between syntax and morphology is not crystal clear in Turkish. This grey area is even more visible with the issue of the Turkish copula verb *i-* (*be*). The verb *i-* has three allomorphs in Turkish: *i-*, *-y*, and zero-marked (-∅). Regardless of the category of its base, the verb *i-* always behaves the same in terms of its stress assignment and the features it can host. Moreover, it is always detachable, meaning that the allomorph *i-* and the two others are in free variation as shown in Example 9a, Example 10a, and Example 11a for verbal bases, and Example 9b, Example 10b, and Example 11b for nominal bases. When the copular verb is detached as a word, the copular verb surfaces as a *i-* (Example 9a, 9b). When both the base and the copular verb surface as a single syntactic word, either *-y* (Example 10a, 10b) or -∅ (Example 11a, 11b) is used. The selection between the -∅ and *-y* is governed by the previous segment; if the previous segment is a consonant -∅ is used, otherwise *-y* is used.



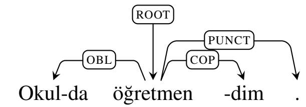
(9) a. Okul-a var-acak i-di-m .
school-DAT reach-FUT be-PST-1SG .
'I was going to reach to school.'



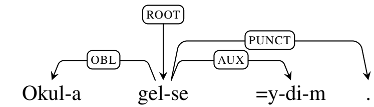
b. Okul-da öğrenci i-di-m .
school-LOC student be-PST-1SG .
'I was a student in the school.'



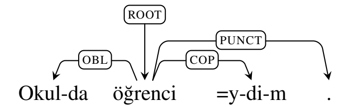
(10) a. Okul-a gel-ecek -ti-m .
school-DAT come-FUT -PST-1SG .
'I was going to come to school.'



b. Okul-da öğretmen -dim .
school-LOC teacher -PST-1SG .
'I was a teacher in the school.'



(11) a. Okul-a gel-se =y-di-m .
school-DAT come-COND =COP-PST-1SG .
'If I went to school.'



b. Okul-da öğrenci =y-di-m .
school-LOC student =COP-PST-1SG .
'I was a student in the school.'

In addition to these characteristics, it also has a clitic-like behaviour when it co-occurs with other clitics such as the question clitic *-mI*. Consider Example 12. When attached, the question clitic comes between the TAM (Tense/Aspect/Modality) marker and the copula.

(12) *Bu kitab-ı oku-yacak mı=y-dı-n?*
this book-ACC read-FUT Q=COP-PST-2SG
'Were you going to read this book?'

Another clue for the clitic status of the copula is its interaction with vowel harmony. When detached, it has its own phonological domain; thus vowel harmony processes do not percolate from the main verb to the copula as seen in Example 9a and Example 10a.

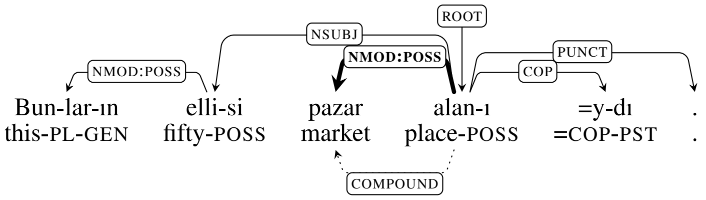
Even though the previous Turkish treebanks are consistent in the decision with regards to the verb *i-* within categories, they lack a unified treatment of the verb *i-* when it surfaces as a clitic as in Example 10a, Example 10b, Example 11a, Example 11b. In previous treebanks, all copular verbs that behave as an `aux` were analyzed within the same word with their host and thus, were not segmented. However, when the copular verb is attached to a nominal base, annotation was made as such that it was analyzed as a separate syntactic unit. After discussing this issue within the UD community,[9] we decided to segment all instances of the verb *i-* as a copula (`cop`) or as an auxiliary verb (`aux`) depending on the category of the base it attaches. With this change, we unified the treatment of all root level clitics which include the question particle *=mı*, focus particles like *=da*, and copular verb particles; thus, followed the UD dependency relations more faithfully. The reason for differentiation between `aux` and `cop` stems from how these syntactic dependencies are defined in the UD framework. The dependency `cop` is specifically used for the copular verb that surfaces with nonverbal predicates as in Example 9b, Example 10b, and Example 11b. In contrast, the dependency `aux` is used when additional TAM elements are expressed by an additional verbal element as in Example 9a, Example 10a, and Example 11a.

[9] For the whole discussion, see `https://github.com/UniversalDependencies/docs/issues/639`

4.2.3 Compound

Another inconsistent annotation in the previous Turkish treebanks was with regards to the compounds and their classifications. The UD framework specifies the use of `compound` as a dependency relation between two heads that have the same syntactic category. Mostly in Turkish PUD, but also in other Turkish treebanks in UD, not only constructions that are formed with two heads, but also some of the constructions that involve genitive-possessive suffixes are marked with the `compound` dependency as in Example 13. We have modified the latter as `nmod:poss`, which is already a convention in use in the UD framework.

(13)



Bun-lar-ın elli-si pazar alan-ı =y-dı .
this-PL-GEN fifty-POSS market place-POSS =COP-PST .

'50 of these were marketplaces.'

Turkish employs different strategies for compounding. These strategies display differences both in their morphological and phonological forms. For our purposes, we divide them into two. Compounds with the compound marker *-(s)I(n)* and compounds without the compound marker *-(s)I(n)*. Some compound types without the compound marker are given in Example 14. These compounds are formed with different types of lexical inputs and can have varying degrees of morpho-phonological strategies, none of which employs a compound marker. We annotated the compounds that do not employ a marker as 'compound'.

(14) a. Noun + Noun
şiş kebap
skewer kebab
'shish kebab'

b. Non-word + Non-word
abur cubur
… …
'junk food'

c. Noun + Non-Word
kitap mitap
book EMPH-book
'book and whatnot'

d. Adverb + Adverb
bugün yarın
today tomorrow
'soon'

e. Verb + Verb
in-di bin-di
get_on-PST get_off-PST
'stopover'

f. Adjective + Adjective
kırık dökük
broken dowdy
'scrap'

The important distinction for our purposes is the existence of a compound marker -*(s)I(n)*. This marker is only observed in Noun+Noun compounds and most of these compounds can be turned into Genitive-Possessive constructions as in Example 15.

(15) a. Noun + Noun
okul bina-sı
school building-3SG
'a school building'

b. Possessive constructions
okul-un bina-sı
school-GEN building-3SG
'the school's building'

We annotated N+N compounds that employ the compound marker *-(s)I(n)* as ‘nmod:poss’. There are two reasons for tagging these compounds with *-(s)I(n)* marker as ‘nmod:poss’. The first one is that the marker is not stable in possessive constructions, it is replaced by the possessive markers. If the possessor is 1SG or 2SG, the marker is replaced with the first person singular possessive *-(I)m* or the second person singular possessive *-(I)n*, respectively. If the possessor is 3SG the marker stays the same. The second reason is plural marking of the compounds. Any plural marking precedes the marker *-(s)I(n)* as opposed to following it, just like in possessive constructions (Example 16).

(16) a. *ders kitab-(lar)-ı*
course book-PL-3SG
‘coursebook(s)’

b. *ders kitab-(lar)-ım*
course book-PL-1SG
‘my coursebook(s)’

c. *ders kitab-(lar)-ın*
course book-PL-2SG
‘your coursebook(s)’

d. *(o-nun) ders kitab-(lar)-ı*
(s/he-GEN) course book-PL-3SG
‘his/her coursebook(s)’

The ability to transition from a compound to genitive-possessive construction, the instability in simple possessive constructions, and the linearization of the plural marker before *-(s)I(n)* makes these word forms more syntactic (compositional) than morphological. This does not mean that word forms with *-(s)I(n)* are not compounds, but within the framework of UD, they are more suited to be classified as ‘nmod:poss’ than ‘compound’.

There is a robust linguistics discussion about the status of the marker *-(s)I(n)* as being classified either as a ‘compound’ or an ‘agreement’ marker. The word forms produced by it are actually referred to as ‘possessive compounds’ (Hayashi 1996; Kunduracı 2013; Taylan and Öztürk Başaran 2014; Öztürk and Taylan 2016), introducing a dilemma even in its own name.

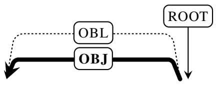
4.2.4 Core Arguments

Turkish poses a problem with regards to the detection of core arguments. hese problems stem from mainly two reasons: core arguments marked with a lexical case and object drop of core arguments. Like Czech, Turkish allows its direct object to be marked with oblique cases. In addition to the structural accusative case, Turkish also makes use of dative, ablative, comitative, and locative on objects, which are the cases that adjuncts can also take. Both the adjunct in Example 18 and the core argument in Example 17 are marked with the same case: COM. When there is no appropriate context that introduces the object earlier, a COM-marked NP becomes obligatory as in Example 17. However, Example 18 is completely fine regardless of context and the existence of COM-marked NP. This is because the COM-marked NP is a core argument in Example 17, whereas it is an adjunct in Example 18.

(17) *Serap *(kız kardeş-i-yle) hep dalga geç-er.*
Serap girl sibling-POSS-COM always make.fun-AOR.
‘Serap makes fun of her sister.’

(18) *Serap okul-a (abla-sı-yla) gid-er.*
Serap school-DAT big.sister-POSS-COM go-AOR
‘Serap goes to school with (her elder sister).’

Turkish can drop its object without any marking on the verb when it is available in the discourse or it is not contradictory within a given context. Since it is impossible to drop the new information or correction in the case of Example 17 without a context that introduces the direct object earlier, we conclude that the NP *kız kardeşiyle* (with her sister) is a core argument. If it were just an adjunct, the phrase can be omittable. Oblique case marking of the core arguments together with the null marking of the contextually available core arguments yields a problem for the annotation process within a framework where the difference between core arguments and non-core arguments is a morphologically-apparent case marking as in the UD framework. Recent discussions in the UD framework also acknowledge this problem (Zeman 2017; Przepiórkowski and Patejuk 2018). They propose a new dependency relation: `obl:arg`. In our annotations, we used the `obj` dependency relation as in Example 19. The UD guidelines state that even though `obj` often carries an accusative case, it may surface with different case markers when the verb dictates a different form, in our case *lexical* case. This approach is also utilized within the most recent Turkish treebank in which they did not distinguish between the objects with accusative case and the objects with non-accusative cases (Kayadelen et al. 2020).



(19) Ütü-den anla-ma-m .
ironing-ABL understand-NEG-1SG .
'I do not know a thing about ironing.'

5 Annotation Tool

Annotation tools are fundamental to the facilitation of the annotation process of many NLP tasks including dependency parsing. UD treebanks are re-annotated or annotated from scratch in line with the annotation guidelines of the UD framework (Nivre et al. 2016). There are several annotation tools that are showcased within the UD framework such as UD Annotatrix (Tyers et al. 2017a) and ConlluEditor (Heinecke 2019). These tools are mostly based on mouse-clicks, and provide graph view and/or text view. Morphological features are, in general, not easy to annotate/edit with the available tools.

We present BoAT, a new annotation tool specifically designed for dependency parsing. To the best of our knowledge, it is the first tool that provide tree view and table view simultaneously. BoAT enables annotators to use both mouse clicks and keyboard shortcuts. In addition, unlike previous dependency parsing annotation tools, which show morphological features as a whole, in BoAT, morphological features are parsed and expanded into multiple columns, as they are one of the most re-annotated fields according to the observations of our annotators. Using BoAT, tokenization can be easily changed, which together with the enhanced presentation of morphological features, are useful properties, especially for agglutinative languages. The tool itself, however, is not specific to agglutinative languages and can be used for other languages as well.

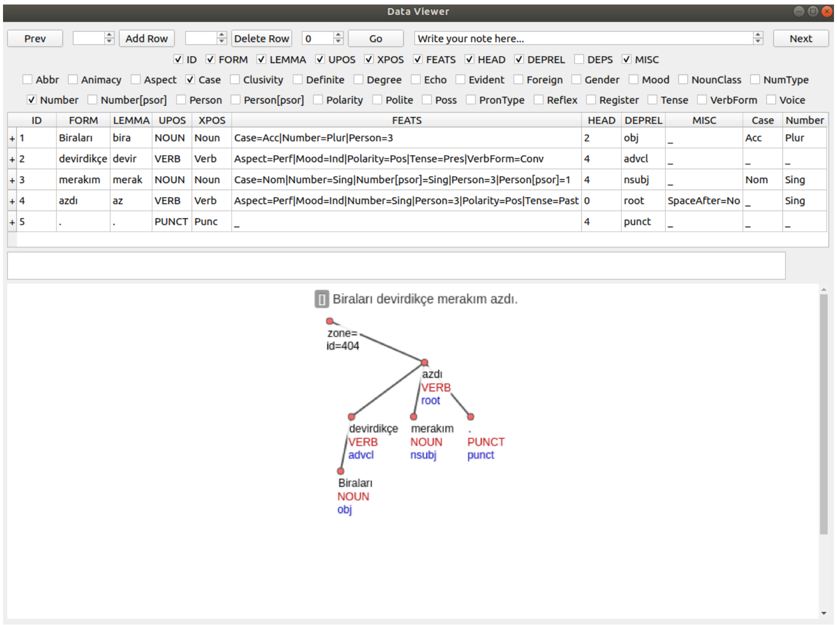
BoAT is designed with the aim of presenting a user-friendly, compact, and practical manual annotation tool that is built upon the preferences of the annotators. It combines useful features from other tools such as changing tokenization, using a validation mechanism, taking notes with novel features such as combining tree and table views, parsing morphological features, and adding keyboard shortcuts to match the needs of the annotators for the dependency parsing task.

While developing BoAT, we received feedback from our annotators in every step of the process. One crucial aspect of annotation is speed. Annotation tools are helpful in this regard but they are still open to advancement in terms of speed. The existing tools within the UD framework mostly rely on mouse clicks and dragging, and the usage of keyboard shortcuts is very limited. Unlike them, almost every possible action within BoAT can be carried out via both mouse clicks and keyboard shortcuts. We aim to decrease the time-wise and ergonomic load introduced by the use of a mouse and to increase speed accordingly.

We also added the note taking option being inspired by BRAT (Stenetorp et al. 2012). While notes are specific to annotations in BRAT, they are specific to each sentence in our tool. This feature enabled our annotators to have better communication and have better reporting power.

5.1 Features

BoAT is a desktop annotation tool which is specifically designed for CoNLL-U files. It offers both tree view and table view as shown in Figure 1 for an example sentence. The upper part of the screen shows the default table view while the lower part shows the tree view. Below we explain briefly the components and some of the properties of the tool.



☑ ID ☑ FORM ☑ LEMMA ☑ UPOS ☑ XPOS ☑ FEATS ☑ HEAD ☑ DEPREL □ DEPS ☑ MISC

□ Abbr □ Animacy □ Aspect ☑ Case □ Clusivity □ Definite □ Degree □ Echo □ Evident □ Foreign □ Gender □ Mood □ NounClass □ NumType

☑ Number □ Number[psor] □ Person □ Person[psor] □ Polarity □ Polite □ Poss □ PronType □ Reflex □ Register □ Tense □ VerbForm □ Voice

	ID	FORM	LEMMA	UPOS	XPOS	FEATS	HEAD	DEPREL	MISC	Case	Number
+	1	Biraları	bira	NOUN	Noun	Case=Acc|Number=Plur|Person=3	2	obj	_	Acc	Plur
+	2	devirdikçe	devir	VERB	Verb	Aspect=Perf|Mood=Ind|Polarity=Pos|Tense=Pres|VerbForm=Conv	4	advcl	_	_	_
+	3	merakım	merak	NOUN	Noun	Case=Nom|Number=Sing|Number[psor]=Sing|Person=3|Person[psor]=1	4	nsubj	_	Nom	Sing
+	4	azdı	az	VERB	Verb	Aspect=Perf|Mood=Ind|Number=Sing|Person=3|Polarity=Pos|Tense=Past	0	root	SpaceAfter=No	_	Sing
+	5	.	.	PUNCT	Punc	_	4	punct	_	_	_

Biraları devirdikçe merakım azdı.

Fig. 1 A screenshot from the tool. The sentence is taken from Example 5.

Tree view: The dependency tree of each sentence is visualized in the form of a graph. Instead of using flat view, hierarchical tree view is used. If you hover the mouse pointer over a token in the tree, the corresponding token in the sentence above the tree is highlighted which gives the user linearly readable tree in order to increase readability and clarity. Tree view is based on the hierarchical view feature in the CoNLL-U Viewer offered by the UD framework.

Table view: Each sentence is shown along with its default fields which are ID, FORM, LEMMA, UPOS, XPOS, FEATS, HEAD, DEPREL, DEPS, and MISC. The morphological features denoted by the FEATS field are parsed into specific subfields for existing morphological features in the UD framework. These subfields are optional in the table view; annotators can choose which subfields they want to see. They are stored in the CoNLL-U file in a concatenated manner.

Customizing table view: Annotators can customize the table view according to their needs by using the checkboxes assigned to the fields and the subfields of the FEATS field shown above the parsed sentence. In this way, the user can organize the table view easily and obtain a clean view by removing the unnecessary fields when annotating. This customization ameliorates readability, and consequently the speed of the annotation. The example in Figure 1 shows a customized table view.

Actions in table view: To ease the annotation process, the most frequently used functions are assigned to keyboard shortcuts. Moreover, annotators can jump to any sentence by simply typing the ID of the sentence. The value in a cell is edited by directly typing when the focus is on that cell. If one of the features is edited, the FEATS cell is updated accordingly.

Changing tokenization: One of the biggest challenges in the annotation process is keeping track of the changes in the segment IDs when new segmentations are introduced. In BoAT, new tokens can be added or existing ones can be deleted to overcome tokenization problems generated during the pre-processing of the text. Moreover, annotating multiword expressions often comes at the cost of updating the segment IDs within a sentence in the case of misdetected multiword expression due to faulty automatic tokenization. Annotators may need an easy way to split a word into two different units. We enabled our annotators to split or join words within our tool by clicking the cells in the first column of the table (written “+” or “-”) or using keyboard shortcuts, which permitted a more accurate analysis of multiword expressions.

Validation: Each tree is validated with respect to the field values before saving the sentence. If an error is detected in the annotated sentence, an error message is issued such as “unknown UPOS value". The error is shown between the table view and the tree view.

Taking notes: With the note feature, the annotator is able to take notes for each sentence as exemplified on the top most line in Figure 1. Each note is attached to the corresponding sentence and stored in a different file with the ID of the sentence.

5.2 Implementation

BoAT[10] is an open-source desktop application. The software is implemented in Python 3 along with PySide2 and regex modules. In addition, CoNLL-U viewer is utilized by adapting some part of the UDAPI library (Popel et al. 2017). Resources consisting of a data folder, the tree view, and validate.py are adopted from the UD-maintained tools[11] for validation check. The data folder is used without any changes while some modifications have been made to validate.py. BoAT is a cross-platform application since it runs on Linux, OS X, and Windows.

The BoAT tool was designed in accordance with the needs of the annotators, and it increases the speed and the consistency of the annotation process on the basis of our annotators’ feedbacks. Currently, BoAT only supports the ConLL-U format of UD since it was

[10] BoAT is available at `https://github.com/boun-tabi/BoAT`

[11] `https://github.com/universaldependencies/tools`

designed specifically for dependency parsing. In the future, it may be extended to support other formats such as ConLL-U Plus format.[12]

6 Experiments

We report the results of our parsing experiments on the BOUN Treebank as well as on its different text types, which collectively serve as a baseline for future studies. In addition to the brand-new BOUN Treebank, we performed parsing experiments on our re-annotated versions of the IMST-UD (Türk et al. 2019a) and PUD (Türk et al. 2019a) treebanks,[13] in order to observe the effect of using additional training and test data.

Most prior studies (Eryiğit et al. 2008; Hall et al. 2007; Durgar El-Kahlout et al. 2014; Sulubacak et al. 2016a,b; Sulubacak and Eryiğit 2018b) on Turkish dependency parsing evaluate the treebanks they use (mostly versions of the IMST-UD Treebank) using MaltParser (Nivre et al. 2007). However, the definition of a well-formed dependency tree for MaltParser is different than the conventions of UD such that the root node may have more than one child in the output of the MaltParser. UD defines a dependency tree with exactly one root node, and it is not possible to have MaltParser produce dependency trees that follow the UD convention. For this reason, we use Stanford's neural parser whose original version achieved the best parsing scores on the IMST-UD Treebank with 69.62 UAS and 62.79 LAS in the CoNLL 2017 Shared Task on Multilingual Dependency Parsing from Raw Text to Universal Dependencies (Zeman et al. 2017), and its modified version (Kanerva et al. 2018) achieved one of the best performances with 70.61 UAS and 64.79 LAS in the follow-up task in 2018 (Zeman et al. 2018). It is currently one of the state-of-the-art dependency parsers. This parser uses unidirectional LSTM modules to generate word embeddings and bidirectional LSTM modules to create possible head-dependency relations. It uses ReLu layers and biaffine classifiers to score these relations. For more information, see Dozat et al. (2017).

As stated in Section 4, the BOUN Treebank consists of 9,761 sentences from five different text types. These text types almost equally contribute to the total number of sentences. For the parsing experiments, we randomly assigned each section to the training, development, and test sets with the 80%, 10%, and 10% percentages, respectively. Table 8 shows the number of sentences in each set of the BOUN Treebank.

Table 8 Division of the BOUN Treebank and its different sections among training, development, and test sets for the experiments.

Treebank	Training set	Development set	Test set	Total
Essays	1,561	196	196	**1,953**
Broadsheet National Newspapers	1,518	190	190	**1,898**
Instructional Texts	1,580	198	198	**1,976**
Popular Culture Articles	1,568	197	197	**1,962**
Biographical Texts	1,576	198	198	**1,972**
BOUN	7,803	979	979	**9,761**

In order to observe the parsing performance for different types of text, we first evaluated the dependency parser for each section separately. Then, we measured the performance of

[12] `https://universaldependencies.org/ext-format.html`

[13] These treebanks are available at `https://github.com/boun-tabi/UD_Turkish-BIMST` and `https://github.com/UniversalDependencies/UD_Turkish-PUD`

the parser on parsing the entire BOUN Treebank. As a final set of experiments, we trained the parser on the training sets of the BOUN Treebank and the re-annotated version of the IMST-UD Treebank separately and together, and tested them on five different settings. With that set of experiments, we aim to measure the difference in performance between the BOUN Treebank and the IMST-UD Treebank and to observe the effect of increasing the training size on performance for the case of Turkish dependency parsing.

In our experiments, we did not perform pre-processing actions such as removing the sentences from the training or test sets that include non-projective[14] As for the pre-trained word vectors used by the dependency parser, we used the Turkish word vectors supplied by the CoNLL-17 organization (Ginter et al. 2017).

For the evaluation of the dependency parser, we used the word-based unlabeled attachment score (UAS) and labeled attachment score (LAS) metrics. UAS is measured as the percentage of words that are attached to the correct head, and LAS is defined as the percentage of words that are attached to the correct head with the correct dependency type. In the experiments, we used gold POS tags instead of automatic predictions of them.

6.1 Parsing Results on the BOUN Treebank

Table 9 shows the parsing results of the test sets for each section in the BOUN Treebank and the BOUN Treebank as a whole in terms of the labeled and unlabeled attachment scores. In these experiments, the parser has been trained by using the training set of the BOUN Treebank.

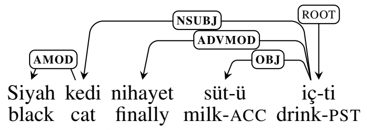
Table 9 UAS and LAS F1 scores of the parser on the BOUN Treebank.

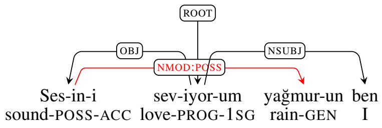
Treebank	**UAS F1-score**	**LAS F1-score**
Essays	68.73	59.18
Broadsheet National Newspapers	**81.59**	**76.04**
Instructional Texts	79.22	72.65
Popular Culture Articles	77.69	71.13
Biographical Texts	80.28	73.68
BOUN Treebank	77.36	70.37

We observed that the highest and lowest LAS were obtained on the *Broadsheet National Newspapers* section and the *Essays* section of the BOUN Treebank, respectively. The parser achieved more or less similar performance on the remaining three sections.

To understand the possible reasons behind the performance differences between the parsing scores of the five sections of the BOUN Treebank, we compared the sections with

[14] In a non-projective sentence, the dependency edges cannot be drawn in the plane above the sentence without any two edges crossing each other, as in Example ii. However, in a projective sentence, the dependency edges can be drawn in this manner with no edges crossing, as in Example i (Nivre 2009).

(i) 
NSUBJ
ROOT
ADVMOD
AMOD
OBJ
Siyah kedi nihayet süt-ü iç-ti
black cat finally milk-ACC drink-PST
‘Black cat finally drank the milk’

(ii) 
ROOT
OBJ
NSUBJ
NMOD:POSS
Ses-in-i sev-iyor-um yağmur-un ben
sound-POSS-ACC love-PROG-1SG rain-GEN I
‘I love the sound of the rain’

dependencies. All sentences in the treebanks were included in the experiments.

respect to the average token count and the average dependency arc length in a sentence. Figure 2 shows these statistics for the five sections of the BOUN Treebank. We observed that both the average token count and the average dependency arc length metrics are the highest in the *Broadsheet National Newspapers* section. The second highest in both metrics is the *Essays* section. The averages for the *Instructional Texts*, *Popular Culture Articles*, and *Biographical Texts* sections are close to each other.

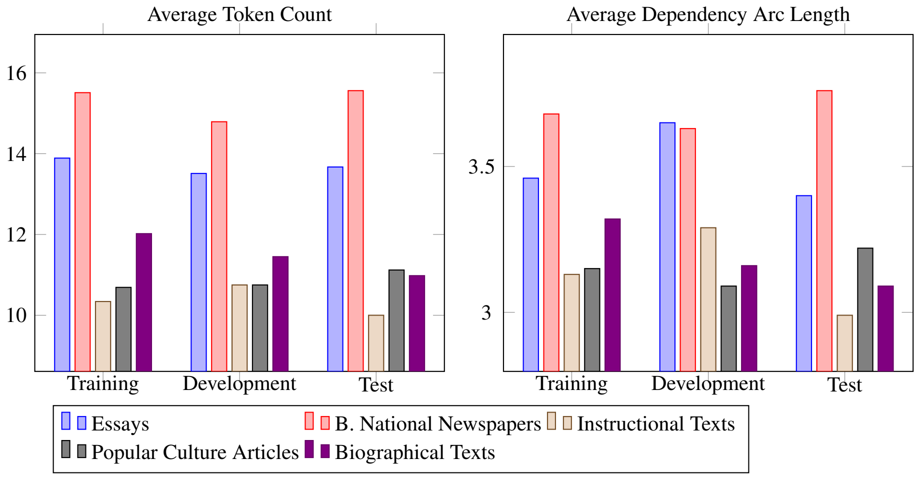


Fig. 2 The average token count and the average dependency arc length in a sentence for the five different sections of the BOUN Treebank.

We anticipate that the higher these two metrics are in a sentence, the harder the task of constructing the dependency tree of that sentence is. In Figure 2, we observe that all of the sections except the *Broadsheet National Newspapers* conform with this hypothesis. However, the *Broadsheet National Newspapers*, which has the highest numbers of these metrics holds the best parsing performance in terms of the UAS and LAS metrics. We believe that this increase in scores is due to the lack of interpersonal differences in writing in journalese and the editorial process behind the journals and magazines.

6.2 Parsing Results on Combinations of Treebanks

In Table 10, we present the success rates of the parser trained and tested on different combinations of the three Turkish treebanks: the BOUN Treebank and the re-annotated versions of the IMST-UD and Turkish PUD treebanks.

The parser is trained separately on the training sets of the IMST-UD and BOUN treebanks, and then, by combining these two training sets (denoted as BOUN+IMST-UD in the first column of Table 10). Originally created for evaluation purposes (Zeman et al. 2017), the PUD Treebank is not used in the training phase of these experiments due to its smaller size compared to the other two treebanks, instead it used as an additional test set in the evaluations.

Five different test sets are provided in the third column of Table 10: the test set of the BOUN Treebank (BOUN), the test set of the IMST-UD treebank (IMST-UD), the Turkish PUD Treebank (PUD), the combined test sets of the BOUN and IMST-UD treebanks

(BOUN+IMST-UD), and the combined test sets of the BOUN and IMST-UD treebanks and the PUD Treebank (BOUN+IMST-UD+PUD).

Each of the trained models is tested on these five test sets. We observe the following:

- The parser model trained on the BOUN Treebank outperforms the one trained on IMST-UD by at least 10% in LAS on the first and third test sets (and ~5% on the fourth and fifth sets). Not surprisingly, the parser trained on IMST-UD performs better on its own test set than the parser model trained on the BOUN Treebank. But the performance difference here is smaller than the one between these two models tested on the BOUN Treebank test set. The parser trained on BOUN outperforms the parser trained on IMST-UD by ~8% in UAS and more than 10% on LAS when tested on the BOUN test set. On the other hand, for the case of the IMST-UD test set, the parser trained on IMST-UD outperforms the parser trained on BOUN by only ~2% in UAS and LAS. Having less amount of training data and a more inconsistent annotation history might be the cause of the inferior performance of the IMST-UD Treebank when compared to the BOUN Treebank.
- Joining the training sets of the BOUN and IMST-UD treebanks improves the parsing performance in terms of attachment scores. The increase in the training size resulted in better parsing scores, contributing to the discussion on the correlation between the size of the corpus and the success rates in parsing experiments (Foth et al. 2014; Ballesteros et al. 2012).
- The worst results by all the models were obtained on the PUD Treebank used as a test set. The different nature of the PUD Treebank compared to the other Turkish treebanks may have an effect on this performance drop. This treebank includes sentences translated from different languages by professional translators and hence, the sentences have different structures than the sentences of the other two treebanks. This difference in structures is a result of the different environments in which these texts are brewed, namely a living corpus (BOUN and IMST-UD) and well-edited translations (PUD).

In order to investigate the differences in the percentages of certain dependency relations between the treebanks used in the experiments, we present the distribution of the dependency relation types across the previous[15] and re-annotated versions of the IMST-UD and PUD treebanks, and the BOUN Treebank in Table 11.

When comparing the BOUN Treebank and the re-annotated version of the IMST-UD Treebank, we observed that the percentages of the `case`, `compound`, and `nmod` types were lower by more than 1% in the BOUN Treebank. The percentage of the `root` type was also lower in the BOUN Treebank by almost 2%, which indicates that the average token count is higher in this treebank with respect to the re-annotated version of the IMST-UD Treebank. However, the percentage of the `nmod:poss` type was higher by more than 2% and the `obl` type was higher by more than 3% in the BOUN Treebank. We believe that these differences are due to the text type we utilized. Unlike IMST-UD, the BOUN Treebank includes essay and autobiography text types. These types make frequent use of postpositional phrases such as *bana göre (in my opinion)* or *1920'ye kadar (until 1920)*, which are encoded with `case` dependency relations. Additionally, the language is less formal compared to the non-fiction and news text types, which are the only registers that the IMST-UD Treebank incorporates. This formality difference explains the lower usage of `compound`.

Moreover, when comparing the BOUN Treebank with the re-annotated version of the Turkish PUD Treebank, we observed that the highest percentage difference was for the `obl`

[15] The UD 2.3 versions of these treebanks are used in the experiments.

Table 10 The performance of the parser on five different test sets according to UAS and LAS metrics. On each test set, the performance of the parser in the following settings is measured: when trained using only the IMST-UD Treebank, when trained using only the BOUN Treebank, and when trained using these two treebanks together.

Training set	**Training size**	**Test set**	**Test size**	**UAS F1-score**	**LAS F1-score**
IMST-UD	3,685	BOUN	979	69.38	58.65
BOUN	7,803	BOUN	979	77.36	70.37
BOUN+IMST-UD	11,488	BOUN	979	77.57	70.50
IMST-UD	3,685	IMST-UD	975	75.49	65.53
BOUN	7,803	IMST-UD	975	73.63	62.92
BOUN+IMST-UD	11,488	IMST-UD	975	76.86	66.79
IMST-UD	3,685	PUD	1,000	65.28	49.50
BOUN	7,803	PUD	1,000	72.33	59.57
BOUN+IMST-UD	11,488	PUD	1,000	72.76	60.39
IMST-UD	3,685	BOUN+IMST-UD	1,954	71.89	61.62
BOUN	7,803	BOUN+IMST-UD	1,954	75.67	66.99
BOUN+IMST-UD	11,488	BOUN+IMST-UD	1,954	77.25	68.82
IMST-UD	3,685	BOUN+IMST-UD+PUD	2,954	69.03	56.37
BOUN	7,803	BOUN+IMST-UD+PUD	2,954	74.22	63.78
BOUN+IMST-UD	11,488	BOUN+IMST-UD+PUD	2,954	75.30	65.17

type which is higher in the BOUN Treebank by more than 7%. This difference is again a result of using different text types. The Turkish PUD Treebank consists of Wikipedia articles in which the adjuncts are expected to be used less than the text types we utilized. The other relation types whose percentages are higher in BOUN by more than 1% were the `root` type which indicates that the average token count is lower in the BOUN Treebank, and the `conj` type indicating that the BOUN Treebank has more conjunct relations which sometimes increased the complexity of a sentence in terms of dependency parsing. These observations suggest that the differences in the parsing scores of these two treebanks might not stem from the varying percentages of the dependency relations; rather, they might stem from the complexity expressed in the text and how well this complexity is handled.

In the comparison of the previous and re-annotated versions of the IMST-UD Treebank with respect to the distribution of dependency relation types, we see that the percentages of the `advmod`, `cc`, `ccomp`, and `nsubj` types increased by approximately 1% in the re-annotated version. In contrast, the percentage of `nmod` is reduced by more than 3% in the re-annotated version. The reason behind this decrease lies in the fact that in the previous version of the treebank, nominalized verbs which behave like a converb (Göksel and Kerslake 2005) are considered nominal modifiers. However, these nominalized verbs actually construct embedded clauses and therefore are treated as clausal modifiers. In addition, the `obl` percentage increased more than 1% in the re-annotated version.

The `vocative` type no longer exists in the re-annotated version and the newly introduced types that are absent in the previous version are the `advcl, advcl:cond, aux, cc:preconj, clf, dislocated, goeswith, iobj, orphan` and `xcomp` relation labels.

When we analyze the differences between the previous and re-annotated versions of the PUD Treebank, we observe that, the biggest difference is in the `compound` relation with a 10% reduction. On the other hand, the biggest increase in the percentage of a relation is in the `nmod:poss` relation with a more than 6% increase in the re-annotated version.

This is because in the previous annotation of the PUD Treebank, some constructions that involve genitive-possessive suffixes are marked with the `compound` dependency label. Such relations have been corrected as `nmod:poss`. Other noteworthy differences are in the `fixed` and `xcomp` relations with a more than 1% decrease and in the `flat`, `nsubj`, and `obl` relations with a more than 1% increase in the re-annotated treebank.

Table 11 Comparison of the previous and re-annotated versions of the IMST-UD and PUD treebanks, and the BOUN Treebank on the distribution of dependency relation labels. The black numbers represent the counts and the gray numbers show their percentages.

Relation type	**IMST-UD** (previous)	**IMST-UD** (re-annotated)	**PUD** (previous)	**PUD** (re-annotated)	**BOUN**
`acl`	1,455 (%2.5)	1,538 (%2.65)	-	515 (%3)	3,494 (%2.85)
`acl:relcl`	-	-	514 (%3.04)	-	-
`advcl`	-	926 (%1.59)	405 (%2.4)	435 (%2.6)	2,595 (%2.12)
`advcl:cond`	-	110 (%0.19)	-	13 (%0.07)	269 (%0.22)
`advmod`	1,872 (%3.2)	2,422 (%4.17)	1,716 (%10.16)	1,624 (%9.6)	5,278 (%4.31)
`advmod:emph`	973 (%1.67)	976 (%1.68)	145 (%0.86)	143 (%0.8)	1,724 (%1.41)
`amod`	3,451 (%5.94)	3,337 (%5.74)	1,224 (%7.25)	1,318 (%7.8)	7,869 (%6.43)
`appos`	51 (%0.09)	136 (%0.23)	36 (%0.21)	166 (%1)	506 (%0.41)
`aux`	-	1 (%0.002)	21 (%0.12)	4 (%0.02)	39 (%0.03)
`aux:q`	209 (%0.36)	211 (%0.36)	-	1 (%0.01)	269 (%0.22)
`case`	2,183 (%3.76)	2,242 (%3.86)	694 (%4.1)	697 (%4.1)	3,290 (%2.69)
`cc`	870 (%1.5)	879 (%3.1)	519 (%3.1)	520 (%3.1)	2,800 (%2.29)
`cc:preconj`	-	3 (%0.005)	8 (%0.05)	8 (%0.05)	134 (%0.11)
`ccomp`	36 (%0.06)	626 (%1.08)	30 (%0.18)	171 (%1)	1,512 (%1.24)
`clf`	-	8 (%0.01)	10 (%0.06)	10 (%0.06)	122 (%0.1)
`compound`	2219 (%3.82)	1,977 (%3.40)	2012 (%11.91)	314 (%1.9)	2,381 (%1.95)
`compound:lvc`	512 (%0.88)	522 (%0.90)	-	186 (%1.1)	1,218 (%1.0)
`compound:redup`	199 (%0.34)	219 (%0.37)	-	9 (%0.05)	457 (%0.37)
`conj`	3,718 (%6.40)	3,529 (%6.07)	640 (%3.79)	696 (%4.1)	7,250 (%5.92)
`cop`	813 (%1.40)	851 (%1.46)	517 (%3.06)	496 (%2.9)	1,289 (%1.05)
`csubj`	7 (%0.01)	82 (%0.14)	115 (%0.68)	93 (%0.5)	546 (%0.45)
`dep`	1 (%0.002)	1 (%0.002)	3 (%0.02)	3 (%0.02)	9 (%0.01)
`det`	2,040 (%3.51)	1,975 (%3.39)	671 (%3.97)	680 (%4)	4,938 (%4.03)
`det:predet`	-	-	10 (%0.06)	8 (%0.05)	-
`discourse`	154 (%0.27)	150 (%0.26)	5 (%0.03)	5 (%0.03)	381 (%0.31)
`dislocated`	-	20 (%0.03)	2 (%0.01)	5 (%0.03)	28 (%0.02)
`fixed`	40 (%0.07)	25 (%0.04)	204 (%1.21)	1 (%0.01)	12 (%0.01)
`flat`	910 (%1.57)	902 (%1.55)	4 (%0.02)	409 (%2.4)	2,039 (%1.67)
`flat:name`	-	-	247 (%1.46)	-	-
`goeswith`	-	3 (%0.005)	1 (%0.01)	1 (%0.01)	4 (%0.002)
`iobj`	-	354 (%0.61)	90 (%0.53)	138 (%0.8)	164 (%0.13)
`list`	-	-	-	-	40 (%0.03)
`mark`	76 (%0.13)	86 (%0.15)	6 (%0.03)	5 (%0.03)	117 (%0.10)
`nmod`	3,780 (%6.51)	1,870 (%3.22)	161 (%0.95)	174 (%1)	1,371 (%1.12)
`nmod:arg`	-	-	110 (%0.65)	-	-
`nmod:poss`	3,534 (%6.08)	3,598 (%6.19)	722 (%4.27)	1,881 (%11)	10,393 (%8.49)
`nsubj`	3,747 (%6.45)	4,430 (%7.63)	1,023 (%6.05)	1,238 (%7.3)	8,499 (%6.94)
`nummod`	621 (%1.07)	567 (%0.98)	207 (%1.22)	263 (%1.6)	1,568 (%1.28)
`obj`	4,307 (%7.41)	3,743 (%6.44)	816 (%4.83)	945 (%5.6)	7,381 (%6.03)
`obl`	4,444 (%7.65)	3,824 (%6.58)	148 (%0.88)	412 (%2.4)	12,015 (%9.82)
`obl:tmod`	-	-	232 (%1.37)	-	-
`orphan`	-	12 (%0.02)	12 (%0.07)	8 (%0.05)	84 (%0.07)
`parataxis`	11 (%0.02)	11 (%0.02)	74 (%0.44)	15 (%0.09)	209 (%0.17)
`punct`	10,228 (%17.61)	10,257 (%17.65)	2,150 (%12.72)	2,148 (%12.7)	20,116 (%16.44)
`root`	5,635 (%9.69)	5,635 (%9.69)	1,000 (%5.91)	1,000 (%5.91)	9,761 (%7.97)
`vocative`	1 (%0.002)	-	1 (%0.001)	-	88 (%0.07)
`xcomp`	-	39 (%0.07)	381 (%2.26)	125 (%0.7)	125 (%0.1)
Total	58,097	58,098	16,886	16,886	122,384

7 Conclusion

In this paper, we presented the largest and the most comprehensive Turkish treebank with 9,761 sentences: the BOUN Treebank. In the treebank, we encoded the surface form of the sentences, universal part of speech tags, lemmas, and morphological features for each segment, as well as the syntactic relations between these segments. We explained our annotation methodology in detail. We present our data online with a history of the changes we applied and our guidelines. We also present an overview of other Turkish treebanks. Moreover, we explained our linguistic decisions and annotation scheme that are based on the UD framework. We provided examples for the challenging issues that are present in the BOUN Treebank as well as other treebanks that we re-annotated.

In addition to such contributions, we provided a description of our annotation tool: BoAT. We explained our motivation for such an initiative in detail. We also provide the tool and the documentation online.

Lastly, we evaluated our new treebank on the task of dependency parsing. We report UAS and LAS F1-scores with regards to specific text types and treebanks. We also showcase the results of the experiments where our new treebank was used with the re-annotated version of the IMST-UD and PUD treebanks. All the tools and materials that are present in the paper are available on our webpage `https://cmpe.boun.edu.tr/boun-pars`.

8 Declarations

8.1 Funding

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant number 117E971 and as BIDEB 2211 graduate scholarship.

8.2 Conflicts of Interest

Not Applicable

8.3 Availability of data and material

Our material regarding our treebank and tool are available online. The links are provided within the text.

8.4 Code availability

Our code regarding R and Python scripts are available online. The links are provided within the text.

Acknowledgements We are immensely grateful to Prof. Yeşim Aksan and the other members of the Turkish National Corpus Team for their tremendous help in providing us with sentences from the Turkish National Corpus. We are also thankful to the anonymous reviewers from SyntaxFest'19 and LAW XIII, as well as to Çağrı Çöltekin for his constructive comments on the re-annotation process of the IMST and PUD Treebanks. GEBIP Award of the Turkish Academy of Sciences (to A.O.) is gratefully acknowledged.

References

Aksan Y., Aksan M., Koltuksuz A., Sezer T., Mersinli Ü., Demirhan U. U., Yılmazer H., Atasoy G., Öz S., Yıldız İ., Kurtoğlu Ö. (2012) Construction of the Turkish National Corpus (TNC). In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, European Language Resources Association (ELRA), Istanbul, Turkey, pp. 3223–3227, URL `http://www.lrec-conf.org/proceedings/lrec2012/pdf/991_Paper.pdf`

Atalay N. B., Oflazer K., Say B. (2003) The annotation process in the Turkish Treebank. In: *Proceedings of 4^{th} International Workshop on Linguistically Interpreted Corpora (LINC-03) at EACL 2003*, URL `https://www.aclweb.org/anthology/W03-2405`

Aygen G. (2003) Extractability and the nominative case feature on tense. In: Özsoy S., Akar D., Nakipoğlu-Demiralp M., Taylan E. E., Aksu-Koç A. (eds.) *Studies in Turkish Linguistics: Proceedings of the 10^{th} International Conference in Turkish Linguistics*, İstanbul

Ballesteros M., Herrera J., Francisco V., Gervás P. (2012) Are the existing training corpora unnecessarily large? *Procesamiento del Lenguaje Natural* (48):21–27

Bickel B., Nichols J. (2013) Inflectional synthesis of the verb. In: Dryer M. S., Haspelmath M. (eds.) *The World Atlas of Language Structures Online*, Max Planck Institute for Evolutionary Anthropology, Leipzig

Brants T. (2000) TnT – a statistical part-of-speech tagger. In: *Sixth Applied Natural Language Processing Conference*, Association for Computational Linguistics, Seattle, Washington, USA, pp. 224–231, DOI 10.3115/974147.974178, URL `https://www.aclweb.org/anthology/A00-1031`

Çetinoğlu Ö., Çöltekin Ç. (2016) Part of speech annotation of a Turkish-German code-switching corpus. In: *Proceedings of the 10^{th} Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, Association for Computational Linguistics, Berlin, Germany, pp. 120–130, DOI 10.18653/v1/W16-1714, URL `https://www.aclweb.org/anthology/W16-1714`

Çetinoğlu Ö. (2009) *A large scale LFG grammar for Turkish*. PhD thesis, Sabanci University

Çöltekin Ç. (2010) A freely available morphological analyzer for Turkish. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, European Language Resources Association (ELRA), Valletta, Malta, URL `http://www.lrec-conf.org/proceedings/lrec2010/pdf/109_Paper.pdf`

Çöltekin Ç. (2015) A grammar-book treebank of Turkish. In: Dickinson M., Hinrichs E., Patejuk A., Przepiórkowski A. (eds.) Proceedings of the 14^{th} Workshop on Treebanks and Linguistic Theories (TLT 14), pp. 35–49

Çöltekin Ç. (2016) (When) do we need inflectional groups? In: *Proceedings of The 1^{st} International Conference on Turkic Computational Linguistics*

Comrie B., Haspelmath M., Bickel B. (2008) The Leipzig glossing rules: Conventions for interlinear morpheme-by-morpheme glosses. Department of Linguistics of the Max Planck Institute for Evolutionary Anthropology and the Department of Linguistics of the University of Leipzig

Dozat T., Qi P., Manning C. D. (2017) Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* pp. 20–30

Durgar El-Kahlout İ., Akın A. A., Yılmaz E. (2014) Initial explorations in two-phase Turkish dependency parsing by incorporating constituents. In: *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, Dublin City University, Dublin, Ireland, pp. 82–89, URL https://www.aclweb.org/anthology/W14-6108

Eryiğit G., Pamay T. (2007) ITU validation set for Metu-Sabancı Turkish treebank. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi* 7(1):31–37

Eryiğit G., Nivre J., Oflazer K. (2008) Dependency parsing of Turkish. *Computational Linguistics* 34(3):357–389

Foth K. A., Köhn A., Beuck N., Menzel W. (2014) Because size does matter: The Hamburg Dependency Treebank. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, European Language Resources Association (ELRA), Reykjavik, Iceland, pp. 2326–2333, URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/860_Paper.pdf

Ginter F., Hajič J., Luotolahti J., Straka M., Zeman D. (2017) *CoNLL 2017 Shared Task - Automatically Annotated Raw Texts and Word Embeddings*. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University

Göksel A., Kerslake C. (2005) *Turkish: A Comprehensive Grammar*. Comprehensive grammars, Routledge

Hall J., Nilsson J., Nivre J., Eryiğit G., Megyesi B., Nilsson M., Saers M. (2007) Single malt or blended? A study in multilingual parser optimization. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Association for Computational Linguistics, Prague, Czech Republic, pp. 933–939, URL https://www.aclweb.org/anthology/D07-1097

Hayashi T. (1996) The dual status of possessive compounds in modern Turkish. *Symbolae Turcologicae* 6:119–129

Heinecke J. (2019) ConlluEditor: a fully graphical editor for Universal Dependencies treebank files. In: *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, Association for Computational Linguistics, Paris, France, pp. 87–93, DOI 10.18653/v1/W19-8010, URL https://www.aclweb.org/anthology/W19-8010

Hoffman B. (1995) *The computational analysis of the syntax and interpretation of "free" word order in Turkish*. PhD thesis, University of Pennsylvania

İşsever S. (2003) Information structure in Turkish: The word order–prosody interface. *Lingua* 113(11):1025–1053

İşsever S. (2007) Towards a unified account of clause-initial scrambling in Turkish: A feature analysis. *Turkic Languages* 11(1):93–123

Kanerva J., Ginter F., Miekka N., Leino A., Salakoski T. (2018) Turku neural parser pipeline: An end-to-end system for the CoNLL 2018 shared task. In: *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Association for Computational Linguistics, Brussels, Belgium, pp. 133–142, URL http://www.aclweb.org/anthology/K18-2013

Kapan A. (2019) *Derivational networks of nouns and adjectives in Turkish*. Master's thesis, Boğaziçi University, İstanbul, Turkey

Kayadelen T., Öztürel A., Bohnet B. (2020) A gold standard dependency treebank for Turkish. In: *Proceedings of The 12th Language Resources and Evaluation Conference*, European Language Resources Association, Marseille, France, pp. 5156–5163, URL https://www.aclweb.org/anthology/2020.lrec-1.634

Kornfilt J. (1984) *Case marking, agreement, and empty categories in Turkish*. Harvard University

Kornfilt J. (2005) Asymmetries between pre-verbal and post-verbal scrambling in Turkish. The free word order phenomenon: Its syntactic sources and diversity pp. 163–180

Kunduracı A. (2013) *Turkish noun-noun compounds: A process-based paradigmatic account*. PhD thesis, University of Calgary

Kural M. (1992) *Properties of scrambling in Turkish*. Ms, UCLA

Kural M. (1997) Postverbal constituents in Turkish and the linear correspondence axiom. Linguistic Inquiry pp. 498–519

Leech G., Garside R. (1991) Running a grammar factory: The production of syntactically analysed corpora or treebanks. *English Computer Corpora: Selected Papers and Research Guide* pp. 15–32

Maamouri M., Bies A., Buckwalter T., Mekki W. (2004) The Penn Arabic treebank: Building a large-scale annotated Arabic corpus. In: *NEMLAR conference on Arabic language resources and tools*, Cairo, vol. 27, pp. 466–467

Marcus M. P., Santorini B., Marcinkiewicz M. A. (1993) Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19(2):313–330

Megyesi B. (2002) *Data-driven syntactic analysis — Methods and applications for Swedish*. PhD thesis, KTH

Megyesi B., Dahlqvist B., Pettersson E., Nivre J. (2008) Swedish-Turkish parallel treebank. In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, European Language Resources Association (ELRA), Marrakech, Morocco, URL `http://www.lrec-conf.org/proceedings/lrec2008/pdf/121_paper.pdf`

Megyesi B., Dahlqvist B., Csató É. Á., Nivre J. (2010) The English-Swedish-Turkish parallel treebank. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, European Language Resources Association (ELRA), Valletta, Malta, URL `http://www.lrec-conf.org/proceedings/lrec2010/pdf/116_Paper.pdf`

Nivre J. (2009) Non-projective dependency parsing in expected linear time. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 351–359

Nivre J., Nilsson J., Hall J. (2006) Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, European Language Resources Association (ELRA), Genoa, Italy, URL `http://www.lrec-conf.org/proceedings/lrec2006/pdf/223_pdf.pdf`

Nivre J., Hall J., Nilsson J., Chanev A., Eryiğit G., Kübler S., Marinov S., Marsi E. (2007) Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering* 13(2):95–135

Nivre J., de Marneffe M.-C., Ginter F., Goldberg Y., Hajič J., Manning C. D., McDonald R., Petrov S., Pyysalo S., Silveira N., Tsarfaty R., Zeman D. (2016) Universal Dependencies v1: A multilingual treebank collection. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, European Language Resources Association (ELRA), Portorož, Slovenia, pp. 1659–1666, URL `https://www.aclweb.org/anthology/L16-1262`

Nivre J., Zeman D., Ginter F., Tyers F. M. (2017) *Tutorial on Universal Dependencies*. URL `http://universaldependencies.org/eacl17tutorial/applications.pdf`, Presented at European Chapter of the Association for Computational Linguistics, Valencia [Accessed: 2019 04 08]

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

Oflazer K., Say B., Hakkani-Tür D. Z., Tür G. (2003) Building a Turkish Treebank, Springer Netherlands, Dordrecht, pp. 261–277. DOI 10.1007/978-94-010-0201-1_15

Özsoy A. S. (1988) Null subject parameter and Turkish. In: *Studies on modern Turkish: Proceedings of the 3rd conference on Turkish linguistics*, Tilburg University Press, Tilburg, the Netherlands, pp. 82–90

Özsoy A. S. (2019) *Word Order in Turkish*, vol. 97. Springer

Öztürk B. (2006) Null arguments and case-driven agree in Turkish. In: Boeckx C. (ed.) *Minimalist essays*, John Benjamins Publishing Company, pp. 268–287

Öztürk B. (2008) Non-configurationality: Free word order and argument drop in Turkish. *The Limits of Syntactic Variation Amsterdam: John Benjamins Publishing Company* pp. 411–440

Öztürk B. (2013) Postverbal constituents in SOV languages. Theoretical Approaches to Disharmonic Word Orders pp. 270–305

Öztürk B., Taylan E. E. (2016) Possessive constructions in Turkish. *Lingua* 182:88–108, DOI 10.1016/j.lingua.2015.08.008

Pamay T., Sulubacak U., Torunoğlu-Selamet D., Eryiğit G. (2015) The annotation process of the ITU web treebank. In: *Proceedings of The 9th Linguistic Annotation Workshop*, Association for Computational Linguistics, Denver, Colorado, USA, pp. 95–101, DOI 10.3115/v1/W15-1610, URL `https://www.aclweb.org/anthology/W15-1610`

Popel M., Žabokrtský Z., Vojtek M. (2017) Udapi: Universal API for Universal Dependencies. In: *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, Association for Computational Linguistics, Gothenburg, Sweden, pp. 96–101

Przepiórkowski A., Patejuk A. (2018) Arguments and adjuncts in Universal Dependencies. In: *Proceedings of the 27th International Conference on Computational Linguistics*, Association for Computational Linguistics, Santa Fe, New Mexico, USA, pp. 3837–3852, URL `https://www.aclweb.org/anthology/C18-1324`

Sak H., Güngör T., Saraçlar M. (2008) Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In: *International Conference on Natural Language Processing*, Springer, pp. 417–427

Sak H., Güngör T., Saraçlar M. (2011) Resources for Turkish morphological processing. *Language Resources and Evaluation* 45(2):249–261

Sampson G. (1995) *English for the computer: The SUSANNE corpus and analytic scheme*

Say B., Zeyrek D., Oflazer K., Özge U. (2002) Development of a corpus and a treebank for present-day written Turkish. In: *Proceedings of the 11th International Conference of Turkish Linguistics*, Eastern Mediterranean University, pp. 183–192

Slobin D. I., Bever T. G. (1982) Children use canonical sentence schemas: A crosslinguistic study of word order and inflections. *Cognition* 12(3):229–265

Stenetorp P., Pyysalo S., Topić G., Ohta T., Ananiadou S., Tsujii J. (2012) Brat: A web-based tool for NLP-assisted text annotation. In: *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Avignon, France, pp. 102–107

Sulger S., Butt M., King T. H., Meurer P., Laczkó T., Rákosi G., Dione C. B., Dyvik H., Rosén V., De Smedt K., Patejuk A., Çetinoğlu Ö., Arka I. W., Mistica M. (2013) ParGramBank: The ParGram parallel treebank. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Sofia, Bulgaria, pp. 550–560, URL `https://www.aclweb.org/anthology/P13-1054`

Sulubacak U., Eryiğit G. (2018a) Implementing Universal Dependency, morphology, and multiword expression annotation standards for Turkish language processing. *Turkish*

Journal of Electrical Engineering and Computer Sciences 26:1662–1672, DOI 10.3906/elk-1706-81

Sulubacak U., Eryiğit G. (2018b) Implementing universal dependency, morphology, and multiword expression annotation standards for Turkish language processing. *Turkish Journal of Electrical Engineering & Computer Sciences* 26(3):1662–1672

Sulubacak U., Eryiğit G., Pamay T. (2016a) IMST: A revisited Turkish dependency treebank. In: *Proceedings of TurCLing 2016, the 1st International Conference on Turkic Computational Linguistics*, Ege University Press

Sulubacak U., Gökırmak M., Tyers F., Çöltekin Ç., Nivre J., Eryiğit G. (2016b) Universal Dependencies for Turkish. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, The COLING 2016 Organizing Committee, Osaka, Japan, pp. 3444–3454

Taylan E. E. (1984) *The Function of Word Order in Turkish Grammar*. University of California Press, DOI 10.2307/415636

Taylan E. E. (1986) Pronominal versus zero representation of anaphora in Turkish. In: *Studies in Turkish Linguistics*, John Benjamins, p. 209

Taylan E. E. (2015) *The Phonology and Morphology of Turkish*. Boğaziçi University

Taylan E. E., Öztürk Başaran B. (2014) The notorious -(s)i(n) in Turkish: Neither an agreement nor a compound marker? *Dilbilim Araştırmaları Dergisi* 2:181–199

Türk U., Atmaca F., Betül Özateş Ş., Öztürk Başaran B., Güngör T., Özgür A. (2019a) Improving the annotations in the Turkish Universal Dependency treebank. In: *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, Association for Computational Linguistics, Paris, France, pp. 108–115, DOI 10.18653/v1/W19-8013, URL `https://www.aclweb.org/anthology/W19-8013`

Türk U., Atmaca F., Özateş Ş. B., Köksal A., Öztürk Başaran B., Güngör T., Özgür A. (2019b) Turkish treebanking: Unifying and constructing efforts. In: *Proceedings of the 13th Linguistic Annotation Workshop*, Association for Computational Linguistics, Florence, Italy, pp. 166–177, DOI 10.18653/v1/W19-4019, URL `https://www.aclweb.org/anthology/W19-4019`

Tyers F. M., Sheyanova M., Washington J. N. (2017a) UD annotatrix: An annotation tool for Universal Dependencies. In: *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, Prague, Czech Republic, pp. 10–17

Tyers F. M., Washington J., Çöltekin Ç., Makazhanov A. (2017b) An assessment of Universal Dependency annotation guidelines for Turkic languages. In: *Proceedings of the 5th International Conference on Turkic Languages Processing (TurkLang 2017)*, Tatarstan Academy of Sciences

Xue N., Xia F., Chiou F.-D., Palmer M. (2005) The Penn Chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering* 11(2):207–238

Yuret D., Türe F. (2006) Learning morphological disambiguation rules for Turkish. In: *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, Association for Computational Linguistics, pp. 328–334

Zeman D. (2017) Core arguments in Universal Dependencies. In: *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, Linköping University Electronic Press, Pisa,Italy, pp. 287–296, URL `https://www.aclweb.org/anthology/W17-6532`

Zeman D., Popel M., Straka M., Hajic J., Nivre J., Ginter F., Luotolahti J., Pyysalo S., Petrov S. (2017) CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In: *Proceedings of the CoNLL 2017 Shared Task: Multilingual*

Parsing from Raw Text to Universal Dependencies, Association for Computational Linguistics, Vancouver, Canada, pp. 1–19, URL `http://www.aclweb.org/anthology/K/K17/K17-3001.pdf`

Zeman D., Hajič J., Popel M., Potthast M., Straka M., Ginter F., Nivre J., Petrov S. (2018) CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In: *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Association for Computational Linguistics, Brussels, Belgium, pp. 1–21, URL `http://www.aclweb.org/anthology/K18-2001`

A Morphological Conversion

Table 12 Mappings of morphological features from the notation of Sak et al. (2011) to the features used in the UD framework.

Sak et al.	UD	Sak et al.	UD
A1sg	Number=Sing| Person=1	ByDoingSo	VerbForm=Conv| Mood=Imp
A2sg	Number=Sing| Person=2	Pos	Polarity=Pos
A3sg	Number=Sing| Person=3	Neg	Polarity=Neg
A1pl	Number=Plur| Person=1	Past	Aspect=Perf| Tense=Past| Evident=Fh
A2pl	Number=Plur| Person=2	Narr	Tense=Past| Evident=Nfh
A3pl	Number=Plur| Person=3	Fut	Tense=Fut| Aspect=Imp
P1sg	Number[psor]=Sing| Person[psor]=1	Aor	Tense=Aor| Aspect=Hab
P2sg	Number[psor]=Sing| Person[psor]=2	Pres	Tense=Pres| Aspect=Imp
P3sg	Number[psor]=Sing| Person[psor]=3	Desr	Mood=Des
P1pl	Number[psor]=Plur| Person[psor]=1	Cond	Mood=Cnd
P2pl	Number[psor]=Plur| Person[psor]=2	Neces	Mood=Nec
P3pl	Number[psor]=Plur| Person[psor]=3	Opt	Mood=Opt
Abl	Case=Abl	Imp	Mood=Imp
Acc	Case=Acc	Prog1	Aspect=Prog| Tense=Pres
Dat	Case=Dat	Prog2	Aspect=Prog| Tense=Pres
Equ	Case=Equ	DemonsP	PronType=Dem
Gen	Case=Gen	QuesP	PronType=Ind
Ins	Case=Ins	ReflexP	PronType=Prs| Reflex=Yes
Loc	Case=Loc	PersP	PronType=Prs
Nom	Case=Nom	QuantP	PronType=Ind
Pass	Voice=Pass	Card	NumType=Card
Caus	Voice=Cau	Ord	NumType=Ord
Reflex	Voice=Rfl	Distrib	NumType=Dist
Recip	Voice=Rcp	Ratio	NumType=Frac
Able	Mood=Abil	Range	NumType=Range
Repeat	Mood=Iter	Inf	VerbForm=Vnoun
Hastily	Mood=Rapid	FutPart	VerbForm=Part| Tense=Future| Aspect=Imp
Almost	Mood=Pro	PastPart	VerbForm=Part| Tense=Past| Aspect=Perf
Stay	Mood=Dur	PresPart	VerbForm=Part| Tense=Pres
While	VerbForm=Conv| Mood=Imp		

B Word Order Statistics of the BOUN Treebank

Table 13 Word order counts and relative percentages of main arguments within the BOUN Treebank when there is no null argument.

Order	Count	Percentage (%)
SOV	1456	59.53
OVS	549	22.44
VSO	165	6.75
SVO	144	5.89
OSV	109	4.46
VOS	23	0.94

Table 14 Word order counts and percentages of main arguments within the BOUN Treebank.

Order	Count	Percentage (%)
OV	5744	37.21
SV	5416	35.09
SOV	1456	9.43
VS	1116	7.23
VO	714	4.63
OVS	549	3.56
VSO	165	1.07
SVO	144	0.93
OSV	109	0.71
VOS	23	0.15

C TNC Registers

Table 15 TNC Details.

ID	Section Name	Number of Words	%	Documents (total)	%
1	Academic prose: Medicine	714,46	1.44%	145	2.91%
2	Academic prose: Social, behavioral sciences	2,892,961	5.83%	432	8.66%
3	Academic prose: Humanities/Arts	2,604,645	5.24%	354	7.09%
4	Academic prose: Natural sciences	1,236,958	2.49%	251	5.03%
5	Academic prose: Politics, law, education	3,857,971	7.77%	587	11.76%
6	Academic prose: Technology, computing, engineering	1,653,909	3.33%	251	5.03%
7	Administrative and regulatory texts, in house use	155,054	0.31%	11	0.22%
8	Print Advertisements	22,311	0.04%	164	3.29%
9	Biographies/Autobiographies	2,372,093	4.78%	158	3.17%
10	Commerce&Finance/Economics	2,282,709	4.6%	120	2.4%
11	E-mail	31,316	0.06%	261	5.23%
12	School essays	56,545	0.11%	10	0.2%
13	Essay	494,747	1%	99	1.98%
14	Excerpts from modern drama scripts	655,618	1.32%	63	1.26%
15	Single and multiple author collections of poems	279,984	0.56%	35	0.7%
16	Novels/short stories	8,271,257	16.65%	566	11.34%
17	Official/govermental documents/leaflets company annual reports etc.; excludes Hansard	594,45	1.2%	56	1.12%
18	Instructional texts	305,829	0.62%	29	0.58%
19	Personal letters	105,693	0.21%	4	0.08%
20	Professional/business letters	20,092	0.04%	1	0.02%
21	Miscellaneous texts	1,932,821	3.89%	108	2.16%
22	Broadsheet national newspapers: arts/cultural material	573,701	1.16%	43	0.86%
23	Broadsheet national newspapers: commerce & finance	759,85	1.53%	67	1.34%
24	Broadsheet national newspapers: miscellaneous material	545,078	1.1%	56	1.12%
25	Broadsheet national newspapers: science material	378,432	0.76%	23	0.46%
26	Broadsheet national newspapers: material on lifestyle leisure belief & thought	1,600,828	3.22%	114	2.28%
27	Broadsheet national newspapers: sports material	662,518	1.33%	57	1.14%
28	Broadsheet national newspapers: column	418,734	0.84%	109	2.18%
29	Non-academic: medical/health matters	99,878	0.2%	5	0.1%
30	Non-academic: social & behavioural sciences	2,411,122	4.85%	87	1.74%
31	Non-academic/non-fiction: humanities&arts	2,644,260	5.32%	156	3.13%
32	Non-academic: natural sciences	93,182	0.19%	7	0.14%
33	Non-academic: politics law education	4,934,042	9.93%	247	4.95%
34	Non-academic: technology, computing, engineering	235,169	0.47%	9	0.18%
35	Popular magazines	667,094	1.34%	48	0.96%
36	Religious texts	975,833	1.96%	46	0.92%
37	Planned speech, whether dialogue or monologue	455,194	0.92%	24	0.48%
38	Forum	468,038	0.94%	68	1.36%
39	Blog	1,199,927	2.42%	119	2.38%
	Total	49,664,303		4990	