Lang Resources & Evaluation (2022) 56:259–307
https://doi.org/10.1007/s10579-021-09558-0

PROJECT NOTES

Resources for Turkish dependency parsing: introducing the BOUN Treebank and the BoAT annotation tool

Utku Türk[1] · Furkan Atmaca[1] · Şaziye Betül Özateş[2] · Gözde Berk[2] · Seyyit Talha Bedir[1] · Abdullatif Köksal[2] · Balkız Öztürk Başaran[1] · Tunga Güngör[2] · Arzucan Özgür[2]

Accepted: 6 August 2021 / Published online: 8 November 2021

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 that 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

✉ Utku Türk
utku.turk@boun.edu.tr

Furkan Atmaca
furkanatmaca@outlook.com.tr

Şaziye Betül Özateş
saziye.bilgin@boun.edu.tr

Gözde Berk
gozde.berk@boun.edu.tr

Seyyit Talha Bedir
talha.bedir@boun.edu.tr

Abdullatif Köksal
abdullatif.koksal@boun.edu.tr

Balkız Öztürk Başaran
balkiz.ozturk@boun.edu.tr

Tunga Güngör
gungort@boun.edu.tr

Arzucan Özgür
arzucan.ozgur@boun.edu.tr

[1] Deparment of Linguistics, Boğaziçi University, İstanbul, Turkey

[2] Department of Computer Engineering, Boğaziçi University, İstanbul, Turkey

efforts for unifying the Turkish UD treebanks through manual re-annotation. To the best of our knowledge, the BOUN Treebank is the largest Turkish UD treebank. It contains a total of 9761 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), and Sampson (1995). These treebanks paved the way for today's ever-growing NLP framework, consisting of NLP applications, treebanks, and tools. Even though the value of a treebank cannot be judged solely by its number of sentences, previous research has shown that the size of a treebank may affect its utility in downstream NLP tasks (Foth et al., 2014). Among the many languages with a growing treebank inventory, Turkish has been one of the less fortunate languages. The latest version[1] of the Turkish IMST-UD Treebank is currently ranked as the 76th treebank out of 183 treebanks in terms of the number of annotated sentences in the Universal Dependencies (UD) project (Nivre et al., 2016). As of the UD version 2.7, the UD project includes 183 treebanks and the largest of them, the UD German-HDT Treebank, consists of 190,000 sentences (Borges Völker et al., 2019). Turkish has posed an enormous challenge for NLP studies due to its complex network of inflectional and derivational morphology, as well as its highly flexible word order. 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 a different 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 9761 sentences

[1] UD version 2.7. Available at http://hdl.handle.net/11234/1-3424.

extracted from five different text types in TNC, 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 the annotation of Turkish data, including the challenges that the copular clitic, embedded constructions, compounds, and lexical cases pose. Turkish treebanking studies present an inconsistent picture in the annotation of such constructions, even though these linguistic phenomena are observed and studied extensively within Turkish linguistic studies.

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 effect on 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 Sect. 2, we briefly explain the morphological and syntactic properties of Turkish. In Sect. 3, we present an extensive review of previous treebanking efforts in Turkish and locate them with regards to each other in terms of their use and their aim. In Sect. 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 Sect. 5. In Sect. 6, we report our experiments and their results. In Sect. 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. This number is about twice of the average of the maximum number of inflectional categories in the other 145 languages covered in Bickel and Nichols (2013). The number of morphological categories increases further when considering derivational processes. Kapan (2019) states that

Table 1 Possible morphological analyses of the word *alın* from Sak et al. (2008)

Root	Category of the root	Features	Gloss	Translation
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!’

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

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 number 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.

With respect to syntactic properties, Turkish has a relatively free word order, which is constrained by discourse elements and information structure (Hoffman, 1995; İşsever, 2003; Kornfilt, 2005; Kural, 1997; Özsoy, 2019; Öztürk, 2008, 2013; Taylan, 1986). 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 Tables 13 and 14 in Appendix 2. 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 is always in pre-verbal position, whereas old information may surface both in pre-verbal 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 pre-verbal 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 argument in a sentence needs to host a case according to its syntactic role, semantic contribution, or the lexical selection of the

[2] Conventions used in the paper are as follows: 1 = first person, 2 = second person, 3 = third person, ABL = ablative, ACC = accusative, AOR = aorist, CAUS = causative, CL = classifier, 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. The dash symbol (-) in linguistics examples marks morpheme boundary, the equal sign (=) is used when the morpheme attached to a base is a clitic. The tilde ∼ is used to indicate partial replication. The asterisk * at the beginning of a sentence shows the sentence's ungrammaticality, and the percentage symbol (%) shows the marginal acceptability of the sentence. Additionally, we presented the analytic words within a box when they are segmented for annotation.

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 & Kerslake, 2005). With certain nominalizer suffixes, the embedded sentences may function as an adverbial, an adjectival, or a nominal.

3 Previous Turkish treebank initiatives

The initial groundwork for Turkish treebanks was laid in Atalay et al. (2003) and Oflazer et al. (2003) following the studies on treebanks for languages such as English, German, Dutch, and many more (Brants et al., 2002; Leech & Garside, 1991; Marcus et al., 1993; Sampson, 1995; van der Beek et al., 2002). The first of its kind, the METU-Sabancı Treebank (MST) consists of 5635 sentences, a subset of the METU corpus that reportedly 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 2007 Shared Task (Eryiğit & 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. Consequently, 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 part of speech types, IMST-UD upped these numbers to 29, 66, and 14, respectively. Yet, the erroneous dependency tagging resulting from morpho-phonological syncretisms lingered long after the publication of the treebank. Moreover, no post-editing effort has been reported. There have been four updates since the first release of the IMST-UD treebank, but there are still mistakes that can 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. All the 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 morphological parser in Oflazer (1994), and its accuracy was enhanced through the morphological disambiguator proposed in Yuret and Türe (2006). The Turkish and Swedish treebanks were annotated using the MaltParser (Nivre et al., 2007) that was 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 1000 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 were included such as the usage of punctuations 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 & Eryiğit, 2018). 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 & Çöltekin, 2019) is another treebank, in which they did not use formal texts. The Turkish-German Code-Switching Treebank consists of bilingual conversation transcriptions as well as their morphological and syntactic annotation. This treebank includes 48 unique conversations and 2184 Turkish-German bilingual sentences that have been annotated with respect to the language in use.

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 2803 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, 2019b). 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), despite the number of unique dependency tags increasing from 31 to 40 with the newly included dependency types (Türk et al., 2019b). 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 were 78.70 UAS and 70.01 LAS (Türk et al., 2019a). We want to note that, we used 5-fold cross validation for the evaluation of the PUD Treebank due to its small size. In each fold, the parser had only 600 sentences for training, and 200 sentences were used as the 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. In Table 2, we present basic statistics about the BOUN Treebank and compare it to the previous monolingual Turkish treebanks. If both UD and non-UD versions are available for a treebank, we only included the UD version in the table.[3]

[3] UD version number of these treebanks is 2.7. Turkish PUD version 2.7 is our re-annotated version.

Table 2 Comparison of the BOUN Treebank to previous monolingual Turkish treebanks

	IMST-UD	IWT-UD	GB	PUD	BOUN
Num. of sentences	5635	5009	2880	1000	9761
Num. of tokens	56,396	44,463	16,803	16,536	121,214
Avg. token count per sentence	10.01	8.88	5.83	16.53	12.41
Avg. dependency arc length	2.71	2.13	1.77	2.91	2.86
Num. of unique POS tags	14	15	16	16	17
Num. of unique features	66	54	79	59	56
Num. of unique dependencies	32	28	41	40	41

4 The BOUN Treebank

In this paper, we introduce a treebank that consists of 9761 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 3 shows the percentages of different domains and media used in TNC.[4]

In our treebank, we included the following text types: essays, broadsheet national newspapers, instructional texts, popular culture articles, and biographical texts. Approximately 2000 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 variations in length, formality, and literary quality. TNC consists of 39 different registers, reported in Table 15 in Appendix 3.[5] The basic statistics for the BOUN Treebank and its different sections are provided in Table 4.

Before the manual annotation of the BOUN Treebank, the sentences were first automatically annotated using an end-to-end 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. Before the annotation process started, we first reviewed the dependency relations in use within the UD framework. Upon reviewing the definitions, we created and annotated a list of unique sentences that we believe are representative of the UD dependency relations in Turkish. Later on, we compared our sentences for certain dependency relations with the examples from already existing Turkish UD treebanks. If our examples and the UD examples were not

[4] Our treebank is available online at https://github.com/UniversalDependencies/UD_Turkish-BOUN/.

[5] This table is retrieved from https://www.tnc.org.tr/about-the-corpus/object/ on September 15, 2020.

Table 3 Composition of the written component of TNC using words as the measurement unit, 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		

Table 4 Sentence and word statistics for the different sections of the BOUN Treebank

Treebank	Number of sentences	Number of tokens	Number of word forms
Essays	1953	27,007	27,557
Broadsheet national newspapers	1898	29,307	29,386
Instructional texts	1976	20,442	20,625
Popular culture articles	1962	21,067	21,263
Biographical texts	1972	23,391	23,553
Total	9761	121,214	122,384

The difference between the number of tokens and words is due to multi-word expressions being represented with a single token, but with multiple words

parallel, we first discussed whether or not our interpretation was correct. We then discussed whether or not there should be any inclusions to the UD guidelines. These discussion were also brought up within the UD community.

After settling on the definitions of the dependency relations, two Turkish native speaker linguists manually annotated the BOUN Treebank using our annotation tool that is presented in Sect. 5. Following the annotation process, two other linguists who did not participate in the manual annotation process 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, we performed a partial double annotation in order to have a consistent annotation scheme before the annotation process of the BOUN Treebank started. For this purpose, the annotators performed an additional annotation task independently for the same set of 1000 randomly selected sentences. The disagreements were discussed and resolved with the entire team of linguists and NLP specialists. The Cohen's Kappa measure of inter-annotator agreement for finding the correct dependency label of the relations is found to be 0.82. The unlabeled and labeled attachment scores between the annotations are 0.83 and 0.75, respectively.

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 disambiguator of Sak et al. (2011). For this purpose, the tokenized sentences were first given to the morphological parser. 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 the 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), which 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 unique tokens excluding punctuations 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

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

mean of precision and recall. Our scores align with the scores reported in the original study of Sak et al. (2011), even though their test set and our set here consist of different text types. While they only used newspaper corpora in the test set, we tested the parser using different text types including broadsheet national newspapers, essays, instructional texts, biographical texts, and popular culture articles.

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 (2018). These features were already stated in the UD guidelines. Table 12 in Appendix 1 shows the automatic conversion from the results of Sak et al. (2011)'s morphological disambiguator. As it is clear from the table, the depth of the morphological representation in Sak et al. (2011) and that in the UD framework do not align perfectly, and there is no one-to-one mapping. For example, an output from Sak et al. (2011) may include both `Narr` and `Past` features. In the automatic conversion, we would end up with `Tense=Past` twice and conflicting values for `Evident` feature. To resolve cases similar to these, we made use of simple rules that detect conflicting features due to our conversion and return appropriate features. Moreover, we used the morphological cues provided by the morphological parser to decide on the UPOS and lemma. All elements of our conversion and post-processing can be found on our Github page.[6]

In our treebank, in addition to the 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.

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 (Aygen, 2003; İşsever 2007; Kural, 1992; Taylan, 1984). With these in mind, we wanted to stick with the conventional

[6] https://github.com/boun-tabi/UD_docs.

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

# sent_id = ins_167

text = Sözü uzatıp seni merakta bıraktım galiba.

# trans = Probably, I beat around the bush and kept you in suspense.

ID	*FORM*	*LEMMA*	*UPOS*	*XPOS*	*FEATS*	*HEAD*	*DEPREL*	*DEPS*	*MISC*
1	Sözü	söz	NOUN	Noun	Case=Acc|Number=Sing|Person=3	2	obj	_	_
2	uzatıp	uza	VERB	Verb	Polarity=Pos|VerbForm=Conv| Voice=Cau	5	advcl	_	_
3	seni	sen	PRON	Pers	Case=Acc|Number=Sing|Person=2	5	obj	_	_
4	merakta	merak	NOUN	Noun	Case=Loc|Number=Sing|Person=3	5	obl	_	_
5	bıraktım	bırak	VERB	Verb	Aspect=Perf|Evident=Fh|Number=Sing| Person=1|Polarity =Pos|VerbForm=Fin| Tense=Past	0	root	_	_
6	galiba	galiba	ADV	Adverb	_	5	advmod	_	SpaceAfter=No
7	.	.	PUNCT	Punc	_	5	punct	_	SpacesAfter=\n

Table 7 The dependeny relation set of the BOUN Treebank

Relation Type	Count	%	Relation type	Count	%
acl	3494	2.85	det	4938	4.03
advcl	2595	2.12	discourse	381	0.31
advcl:cond	269	0.22	dislocated	28	0.02
advmod	5278	4.31	fixed	12	0.01
advmod:emph	1724	1.41	flat	2039	1.67
amod	7,869	6.43	goeswith	4	0.002
appos	506	0.41	iobj	164	0.13
aux	39	0.03	list	40	0.03
aux:q	269	0.22	mark	117	0.10
case	3290	2.69	nmod	1371	1.12
cc	2800	2.29	nmod:poss	10,393	8.49
cc:preconj	134	0.11	nsubj	8499	6.94
ccomp	1512	1.24	nummod	1568	1.28
clf	122	0.1	obj	7,381	6.03
compound	2381	1.95	obl	12,015	9.82
compound:lvc	1218	1.0	orphan	84	0.07
compound:redup	457	0.37	parataxis	209	0.17
conj	7250	5.92	punct	20,116	16.44
cop	1289	1.05	root	9761	7.97
csubj	546	0.45	vocative	88	0.07
dep	9	0.01	xcomp	125	0.01

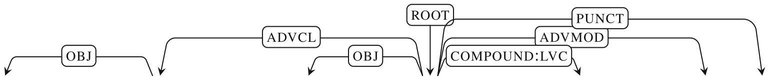
dependency framework and use the recently rising UD framework.[7] One of the main advantages of the UD framework is that it creates directly comparable sets of treebanks with regards to their syntactic representation due to its very nature.

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 the 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, the UD framework differentiates the 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

[7] 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.

expressions, loose syntactic relation, sentential, and extra-sentential.[8] Table 7 shows the dependency relations that we employed in the BOUN 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. In this way, nodes can have more than two children nodes and every node is accessible from the root node. This representation is exemplified in Example 2 using the sentence in Table 6.



(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 Different conventions adopted 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 diverged from these conventions or made the linguistic reasoning behind them more explicit. In this section, we provide the justifications of our linguistic decisions 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, 2019b). Our main concern is to reflect linguistic adequacy in the BOUN Treebank following the Manning's Law (Nivre et al., 2017). During all this work, 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 will first touch upon the issues where we believe the previous conventions in Turkish UD treebanking were erroneous according to UD. These issues include the annotation of the embedded sentences, the treatment of copular verb, the analysis of compounds, and the annotation of classifiers. Next, we will 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 the recently discussed UD alternatives.

4.2.1 Annotation of embedded clauses

The first issue where we diverged from the previous annotation conventions 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 most common embedding strategy in Turkish: nominalization. Due to nominalization, embedded

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

clauses in Turkish can be regarded as nominals since they behave exactly like nominals: They can be marked with an accusative case, can be substituted with any other nominal, and can carry genitive-possessive cases as person marking as shown in Example 3. The embedded clause in the given sentence is shown with square brackets. The whole square bracket can be replaced with a simple noun, like *otobüs* (bus), or a complex noun phrase like *senin otobüsün* (your bus) as in Example 4.

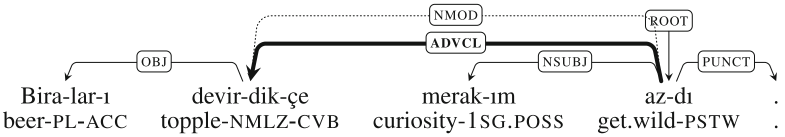
(3) [*Sen-in otobüs-ü sür-düğ-ün*]-*ü gör-dü-m.*
[you-GEN bus-ACC drive-NMLZ-POSS]-ACC see-PST-1SG
'I saw that you drove the bus.'

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

Due to these surface level morphological and syntactic similarities, previous Turkish treebanks in the UD framework, with the exception of the Grammar Book Treebank (Çöltekin, 2015), used dependency relation `obj` instead of `ccomp`, `nsubj` instead of `csubj`, `amod` instead of `acl`, and `advmod` instead of `advcl` to mark the relation of the embedded clause with the matrix verb. In our annotation process, we emphasized the clausal nature of these embedded sentences and their syntactic derivation by focusing on their internal structure reflecting the existence of a temporal domain in the embedded clause. For instance, Example 3 would be unsensical if we had the time adverb *tomorrow* within the embedded clause. This ungramaticality is due to the tense information introduced by the nominalizer '-düğ' in the example sentence. If there were an adverb like *tomorrow* in an embedded clause marked with '-düğ', the previous annotation scheme would not be able to detect the ungrammaticality. However, our annotation scheme is able to detect this ungrammaticality.

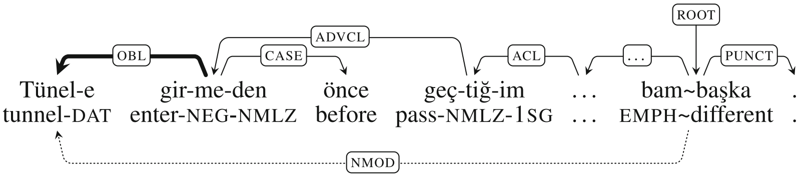
The same argumentation applies to converbs, as well. Converbs are verbal elements of a non-finite adverbial clause (Göksel & Kerslake, 2005). They may act as adverbial adjuncts or as discourse connectives. In the previous annotation processes of Turkish, they were annotated as `nmod`. The reason behind this annotation 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, their ability to host a subject, an object, and a tense/aspect/modality information, we annotated them as `advcl` as in Example 5.[9]

[9] Throughout the paper, changes in the annotation convention introduced by us are shown with bold arcs, whereas the dashed arcs suggest previous annotations. The solid 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.



(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."

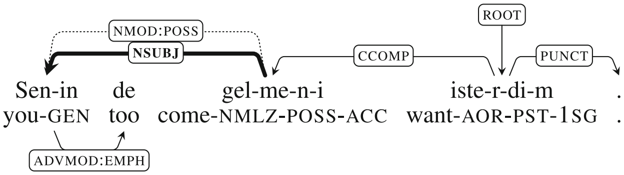
In addition to the annotation of the whole embedded clause, dependents within the embedded clause were erroneously annotated in the previous Turkish annotation schemes. For example, an oblique of an embedded verb used to be attached to the root since the embedded verb is seen as a nominal, and not as 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 & Kerslake, 2005). Despite the morphology, the word *senin* here serves as the subject. Example 8 shows the causativized version of the embedded verb in 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 a direct object. As seen in Examples 7 and 8, the word *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.*
he/she-GEN you-ACC too come.CAUS-NMLZ-POSS-ACC want-AOR-PST-1SG.
‘I would have wanted him/her to bring you, as well.’

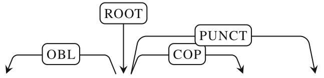
Due to the reasons explained above, in the annotation of embedded clauses we used the dependency relations that emphasize the clausal nature of the nominalized verbs, i.e., `csubj`, `ccomp`, `advcl`, instead of the dependency relations that emphasize the final product of the local derivations, i.e.,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

4.2.2 Copular clitic

One inconsistent issue within the Turkish treebanks was the annotation of the copular clitics. Copular clitics attached to the verbal bases and nominal bases were treated differently although they are essentially the same as we will show below. While the copular clitics on verbal bases were not segmented, the copular clitics on nominal bases were segmented in previous Turkish treebanks. In this section, we will provide our analysis where we segment all copular clitics regardless of their bases.

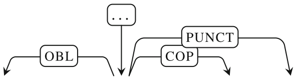
The Turkish copular clitic is the grammaticalized version of the verb "be" which can be indicated as *i-*. This clitic *i-* has three allomorphs in Turkish: (i) analytic *i-*, (ii) suffixal *-y*, and (iii) zero-marked (Ø). The allomorphy of the analytic form is idiosyncratic, meaning the analytic copula form can be used in place the suffixal copula forms most of the time. The analytic form can surface if suffixes *-di* (PST), *-se* (COND), and *-ken* (WHEN or WHILE) come atop a verb that already hosts a TAM (Tense/ Aspect/Modality) marker. The analytic form can also surface in nominal sentences that are marked for tense other than the aorist (*-Ar/Ir*). However, the analytic form cannot surface with the suffix *-mIš* (PRF), except for its use with the aorist as in *yapar imiš*, meaning *he or she used to do*. Example 9a and Example 9b illustrate some examples of the analytic form.

(9) a. analytic COP *i-*



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

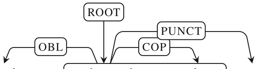
b. analytic COP *i-*



Okul-a var-acak i-ken .
school-DAT reach-FUT be-WHEN .
'I was about arrive to school. ... '

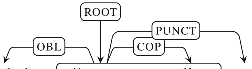
When both the base and the copular verb surface as a single syntactic word indicated with a box in the following examples, either *-y* (Examples 10a, 10b) or ∅ (Examples 11a, 11b) is used. The selection between the ∅ and *-y* is governed by the phonological characteristics of the previous sound; if the previous segment is a consonant ∅ is used, otherwise *-y* is used.

(10) a. zero-marked COP (Ø)



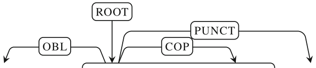
Okul-a gel-ecek =ti-m .
school-DAT come-FUT =PST-1SG .
'I was going to come to school.'

b. zero-marked COP (Ø)



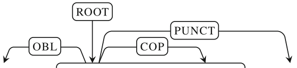
Okul-da öğretmen =di-m .
school-LOC teacher =PST-1SG .
'I was a teacher in the school.'

(11) a. suffixal COP -*y*



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

b. suffixal COP -*y*



Okul-da öğrenci =y-se-m .
school-LOC student =COP-COND-1SG .
'If I was a student in the school...'

What is important for us is that the contribution of these copular clitics is the same for both nominal and verbal bases. In both cases, these copular clitics host the TAM information that cannot be carried by the base (Göksel, 2001). The TAM information itself also does not change according to the category of the stem.

Additionally, the stress patterns of the clitics that attach to nominal and verbal bases are identical. Most of the verbs and common nouns are stressed in the final syllable. When they are marked with a copular clitic, instead of the final syllable which is the copular clitic, the preceding syllable is stressed (Göksel, 2001). This property as well applies regardless of the base the clitic attaches to.

In addition to these characteristics, the copular clitic 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 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.

However, semantic contributions of TAM markers and their interaction with each other provides a counterpoint for segmenting the copular clitic.[10] On a first look, verbs with a copular clitic seem to carry two different tense information. However, two consecutive TAM markers in Turkish do not imply two tenses. While one of them still provides tense information, the other one implies additional aspect. Consider the verb *gelecektim* in Example 13. When either suffix (*-ecek* or *-ti*) is attached to a verb without any additional TAM marker, they mainly provide the tense information. When they are used together as in Example 13, the suffix *-ti* implies the tense information, and the suffix *-ecek* provides the prospective aspect information. This aspect of the copular clitic points towards a solution in which verbs with a copular clitic should be analyzed as a single unit.

[10] We thank the anonymous reviewer for pointing out this issue and initiate this discussion.

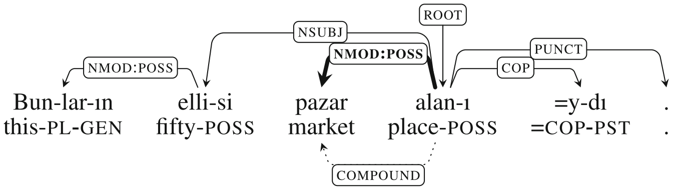
(13) *Okul-a gel-ecek=ti-m ama fikri-m-i değiştir-di-m.*
school-DAT come-FUT=PST-1SG but mind-POSS.1SG-ACC change-PST-1SG
'I was going to come to the school, but I changed my mind.'

After exchanging ideas on this issue within the UD community[11] and considering points mentioned in this section, we decided to segment all instances of the copular verb *i-* as a copula (`cop`). With this change, we unified the treatment of all clitics that may attach to a root which include the question particle *=mı*, focus particles like *=da*, and copular verb particles; thus, followed the UD dependency relations more faithfully.

4.2.3 Compound

Another inconsistent annotation in the previous Turkish treebanks was compounds and their classification. The UD framework suggests that `compound` should be tailored to each language with its particular morphosyntax. Mostly in Turkish PUD, also in other Turkish UD-treebanks, constituents that carry a morphological marker for possessive-compounds are annotated as `compound` like in Example 14. The name 'possessive-compounds' is how the linguistic literature refers to it, but for our purposes we take it as a compositional structure and separate it from the UD dependency 'compound'. This means that our criteria for compound-hood are syntactic composition properties. We have modified cases with the morphological marker *-(s)I(n)* as `nmod:poss`, which is already a convention in use in the UD framework.

(14)



'50 of these were marketplaces.'

Turkish employs different strategies for compounding. These strategies can display differences in their morphological and phonological forms. For our purposes, we divide them into two: (i) compounds with the compound marker *-(s)I(n)* and (ii) compounds without the compound marker *-(s)I(n)*. Some compound types without the compound marker are given in Example 15. These compounds are formed with different types of lexical inputs and can have varying degrees of morpho-phonological properties, none of which employs a compound marker. We annotated the compounds that do not employ a marker as `compound`.

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

(15) 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 the 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 16.

(16) a. Noun + Noun

okul bina-sı
school building-3SG

'school building'

b. Possessive construction

okul-un bina-sı
school-GEN building-3SG

'the school's building'

We annotated Noun+Noun compounds that employ the compound marker *-(s)I(n)* as `nmod:poss`. There are three reasons behind this decision. The first one is that the marker does not survive in possessive constructions, it is replaced by the possessive markers. If the possessor is 1SG or 2SG, the marker is replaced with 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 17). The third reason is that compounds formed with the marker *-(s)I(n)* can have their modifier (non-head) be subject to questions, whereas compounds without it cannot ((Example 18). Questions are considered to be extractions out of syntactic structures which can not target parts of a word form.

(17) a. *ders kitap-(lar)-ı*
course book-PL-3SG
'coursebook(s)'

b. *ders kitap-(lar)-ım*
course book-PL-1SG
'my coursebook(s)'

c. *ders kitap-(lar)-ın*
course book-PL-2SG
'your coursebook(s)'

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

(18) a. i. *adana kebap*
A kebab
'Adana kebab'

ii. **Ne kebap ye-di?*
what kebab eat-PST[3SG]
Intended 'What type of kebab did (s/he) eat?'

b. i. *adana kebab-ı*
A kebab-3SG
'Adana kebab'

ii. *Ne kebab-ı ye-di?*
what kebab-3SG eat-PST[3SG]
'What type of kebab did (s/he) eat?'

As a result, (i) the marker *-(s)I(n)* not surviving possessive constructions and the ability to transition from a compound to genitive-possessive construction shows that the marker *-(s)I(n)* and possessive markers are in a disjunctive blocking relation. This suggests that they are competing for similar grammatical functions. (ii) The plural marker linearizes before the marker *-(s)I(n)*. If *-(s)I(n)* was part of the word form, the plural marking should have linearized to the right of it. This shows that the marker *-(s)I(n)* is not part of the word form. (iii) Parts of the construction formed by *-(s)I(n)* can be targeted by questions. Question formations only target syntactic constituents and not part of word forms. This indicates that structures with *-(s)I(n)* do not constitute an indivisible word form. All these three reasons make constructions involving *-(s)I(n)* more syntactic (compositional) than morphological. This does not unilaterally rule out the constructions with *-(s)I(n)* as 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 as an agreement marker. The word forms produced by it are actually referred to as 'possessive compounds' (Hayashi, 1996; Kunduracı, 2013; Öztürk & Taylan, 2016; Taylan & Öztürk Başaran, 2014), introducing a dilemma even in its own name.

4.2.4 Classifier

The use of the classifier syntactic dependency (`clf`) was also inconsistent within the already existing Turkish UD treebanks. In the UD guidelines, the use of `clf` is limited to languages with highly grammaticized classifier systems. The difference between classifier languages and non-classifier languages is depicted with Chinese (classifier) and English (non-classifier). However, this distinction is not always clear-cut in other languages like Turkish (Sağ, 2019). According to Göksel and Kerslake (2005), numerals can be followed by certain elements such as the enumerator *tane* (piece), measurement denoting words such as *dilim* (slice) and *s̀is̀e* (bottle), and membership/identity denoting words like *örnek* (example) and *kopya* (copy). They show that even though these elements are optional between a numeral and a noun, in partitive constructions with ablative cases, they are obligatorily used. The examples below show that the classifier *tane* (piece) is optional in sentences like Example 19a. However, when the classifier is in inflected form, deleting it makes the sentence ungrammatical as in Example 19b. The sentence becomes marginally acceptable when the inflection is concatenated to the numeral as in Example 19c.

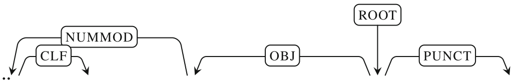
(19) a. *Dört (tane) elma al-dı-m.*
four piece apple buy-PST-1SG
'I bought four apples.'

b. *Küçük-ler-den on *(tane-si) yeter mi?*
small-PL-ABL ten piece-POSS enough Q
'Will ten of the little ones be enough?'

c. % *Küçük-ler-den on-u yeter mi?*
small-PL-ABL ten-POSS enough Q
'Will ten of the little ones be enough?'

Apart from the Turkish PUD Treebank, no previous Turkish treebank has used the `clf` syntactic dependency. In the Turkish PUD Treebank, both measure words and enumerators are annotated using `clf` dependency. As for the other Turkic treebanks, a measure word *bötelke* (bottle) in the Kazakh UD Treebank is annotated using `clf`. On the other hand, in the Uyghur UD Treebank, no `clf` is used. In addition to the UD Treebanks, other recent treebanks such as Kayadelen et al. (2020) that use dependency grammar framework in their annotation, make use of the classifier dependency relation for both enumerators and measurement denoting words.

In the BOUN Treebank and our re-annotated versions of PUD and IMST-UD, we annotated enumerators like *tane* (piece) and *adet* (piece) as classifiers and used the `clf` dependency relation. A slightly modified example sentence from our treebank can be seen in Example 20. One of the UD framework's core ideas is to create a typologically comparable set of treebanks. In this direction, it is important to reflect the use of classifier words in Turkish, even if they are optional.



(20) Üç adet yumurta-yı karış-tır-ın .
three CL egg-ACC mix-CAUS-IMP.2SG .
‘Mix three eggs.’

4.2.5 Core arguments

Turkish also poses a problem with regards to the detection of core arguments. This problem stems from mainly two reasons: core arguments marked with a lexical case and object drop of the 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 22 and the core argument in Example 21 are marked with the same case: COM (comitative). When there is no appropriate context that introduces the object earlier, a COM-marked NP becomes obligatory as in Example 21. However, Example 22 is completely fine regardless of the context and the existence of the COM-marked NP. This is because the COM-marked NP is a core argument in Example 21, whereas it is an adjunct in Example 22.[12]

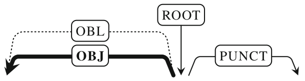
As it can be seen from the examples, Turkish can drop its object without any marking on the verb when it is available in the discourse or it is not contradictory within a given context. Since it is impossible to drop the new information or correction in the case of Example 21 without a context that introduces 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.

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

(22) *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).’

[12] Note that in certain environments where there is an immediate follow-up sentence to Example 21, COM-marked argument can still be omitted as in (i). We thank the anonymous reviewer for pointing this out.
(i) *Serap hep dalga geç-er, ama kardeş-i hiç kız-ma-z-dı.*
Serap always make.fun-AOR but sibling-POSS never get.angry-NEG-AOR.NEG-PST
‘Serap would always make fun of her sister and she would never get angry.’

Oblique case marking of the core arguments together with the optionality 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 morphologi-cally-apparent case marking as in the UD framework. Recent discussions in the UD framework also acknowledge this problem (Przepiórkowski & Patejuk, 2018; Zeman, 2017). They propose a new dependency relation: `obl:arg`. In our annotations, we used the `obj` dependency relation as in Example 23. 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* cases like COM (Example 21) and ABL (Example 23). 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).

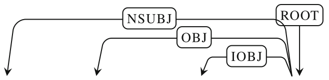


(23) Ütü-den anla-ma-m .
ironing-ABL understand-NEG-1SG .
'I do not know anything about ironing.'

Another core argument specified in the UD guidelines is the `iobj` argument. In their assessment of Turkic treebanks, Tyers et al. (2017b) suggest using case promotion or demotion in passivization or causativization as a clue for determining argumenthood. When sentences are passivized in Turkish, the structural case accusative on the object is deleted in the transformation whereas oblique cases such as the ablative case is not deleted. They use this asymmetry to argue for a non-core analysis of oblique case marked objects. In their proposed annotation scheme, only tokens with non-oblique cases should be annotated as a core argument since only non-oblique cases go through case promotion or demotion. However, as we have previously shown in this section, objects marked with oblique cases behave the same as the objects marked with the accusative cases. Turkish can have oblique cases as a marker of objects even though they do not go through case demotion in passive sentences as in Example 24.

(24) *Ütü-den de anla-n-ma-z mı?*
ironing-ABL EMP understand-PASS-NEG-3SG Q?
'How can one not know anything about ironing?'

Following the reasons specified in this section, we did not make use of case clues in the annotation of `iobj`, instead we utilized the effects born out of context. Following our annotation process, we should annotate the dative marked noun *bana* (to me) using the `iobj` dependency relation if we cannot omit it when the information is already available in the discourse. Without any existing prior context, one cannot omit the dative marked noun in sentences like Example 25 where the main predicate is ditransitive.



(25) Deniz kitab-ı ban-a ver-di .
Deniz book-ACC 1SG-DAT give-PST
'Deniz gave me the book.'

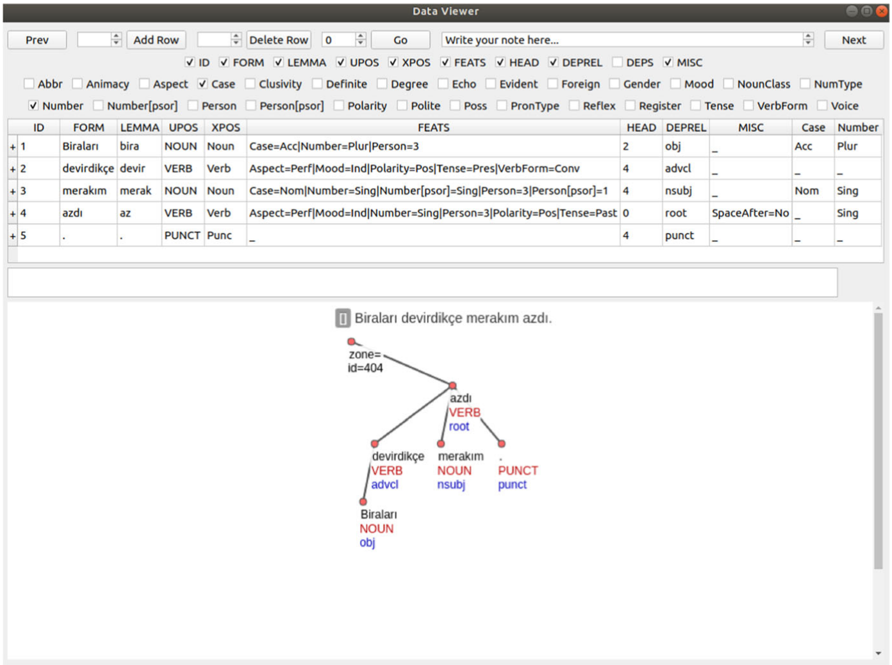
In addition to our treebank, the `iobj` dependency relation is also used in other Turkish and Turkic treebanks. Prior to our re-annotation, the Turkish PUD Treebank already made use of this dependency relation. With our re-annotation, the IMST-UD Treebank also utilizes the `iobj` dependency. The `iobj` relation is also used in a Turkic treebank: the UD Kazakh Treebank (Makazhanov et al., 2015; Tyers et al., 2017b). We believe that the non-optionality of cases like *bana* (to me) in Example 25 and its already existing use in other Turkish and Turkic treebanks justify our usage as well.

4.2.6 Summary of the linguistic considerations

The points made through the linguistic considerations are based on the idea that a language phenomenon needs to be evaluated with regards to its interactions with other phenomena in the same language. There could be opaque processes which require referring to the derivational history of a construction such as nominalization in embeddings, argument dropping (subject, object, indirect object), compound making strategies, or grammatical functions of a clitic. Additionally, a language does not need to employ a structural property uniformly in its grammatical system. Classifiers in Turkish could be an example for this. Example sentences for the UD tagset could already exist in the provided guidelines, but they lack linguistic diagnostics which are crucial to differentiate between the closely related constructions and the mostly opaque processes in a given language. We hope explicitly stating the diagnostics used for an annotation scheme becomes a practice so that the unification process of the treebanks does not follow from standalone examples but rather from testable predictions.

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. There are also annotation tools that have been developed for annotating Turkish treebanks (Atalay et al., 2003; Eryiğit 2007; Pamay et al., 2015; Yıldız et al., 2016). However, they are not specific to the UD



☑ 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

framework. Apart from that, they do not have practical user interfaces regarding dependency parsing.

We present BoAT, a new annotation tool specifically designed for dependency parsing. To the best of our knowledge, it is the first tool that provides 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. The enhanced presentation of morphological features is beneficial for annotators. Using BoAT, tokenization can be easily changed by splitting or joining tokens. This is a useful property, especially for agglutinative languages since they have more suffixes, and tokenization may differ according to the used methods. 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 the tokenization, using a validation mechanism, and 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 Fig. 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.

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 the user hovers the mouse pointer over a token in the tree, the corresponding token in the sentence above the tree is highlighted which gives the user a linearly readable tree in order to increase readability and clarity. The 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. These subfields are a subset of universal and language-specific 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 Fig. 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 expressions 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 permits 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 topmost line in Fig. 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[13] 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 adapted from the UD-maintained tools[14] 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 designed specifically for dependency parsing. In the future, it may be extended to support other formats such as the ConLL-U Plus format.[15]

6 Experiments

We report the results of our parsing experiments on the BOUN Treebank as well as on its different text types, which will 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., 2019b) and PUD (Türk

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

[14] https://github.com/universaldependencies/tools.

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

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	1561	196	196	1953
Broadsheet national newspapers	1518	190	190	1898
Instructional texts	1580	198	198	1976
Popular culture articles	1568	197	197	1962
Biographical texts	1576	198	198	1972
BOUN	7803	979	979	9761

et al., 2019a) treebanks,[16] in order to observe the effect of using additional training and test data.

Most prior studies (Durgar El-Kahlout et al., 2014; Eryiğit et al., 2008; Hall et al., 2007; Sulubacak and Eryiğit 2018; Sulubacak et al., 2016a, 2016b) 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 the Stanford's neural parser whose original version (Dozat et al., 2017) 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 on the same treebank 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 Sect. 4, the BOUN Treebank consists of 9761 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%, respectively. Table 8 shows the number of sentences in each set of the BOUN Treebank.

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 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

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

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

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 data size on performance for 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[17] dependencies. All sentences in the treebanks were included in the experiments. 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).

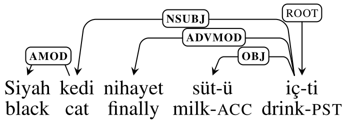
For the evaluation of the dependency parser, we used the unlabeled attachment score (UAS) and labeled attachment score (LAS) metrics. UAS is measured as the percentage of words that are attached to the correct head, and LAS is defined as the percentage of words that are attached to the correct head with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

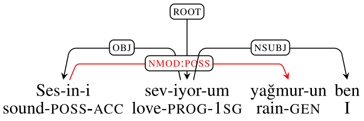
6.1 Parsing results on the BOUN Treebank

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 entire training set of the BOUN Treebank.

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.

[17] 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 (iii). However, in a projective sentence, the dependency edges can be drawn in this manner with no edges crossing, as in (ii) (Nivre, 2009).

(ii)

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'

(iii)

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'

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 respect to the average token count and the average dependency arc length in a sentence. Fig. 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.

Note that, *the average token count* metric, which shows the length of a sentence, and *the average dependency arch length* metric, which depicts the distance between the nodes of the dependency relations in a sentence, can sometimes correlate, although not all long sentences include long range dependencies. We anticipate that the higher these two metrics are in a sentence, the harder the task of constructing the dependency tree of that sentence will be. In Fig. 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 these high scores in this section are 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. We chose to include only these two treebanks that we re-annotated because we wanted to measure the effect of our unification efforts for Turkish treebanking on the parsing accuracy.

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 is 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

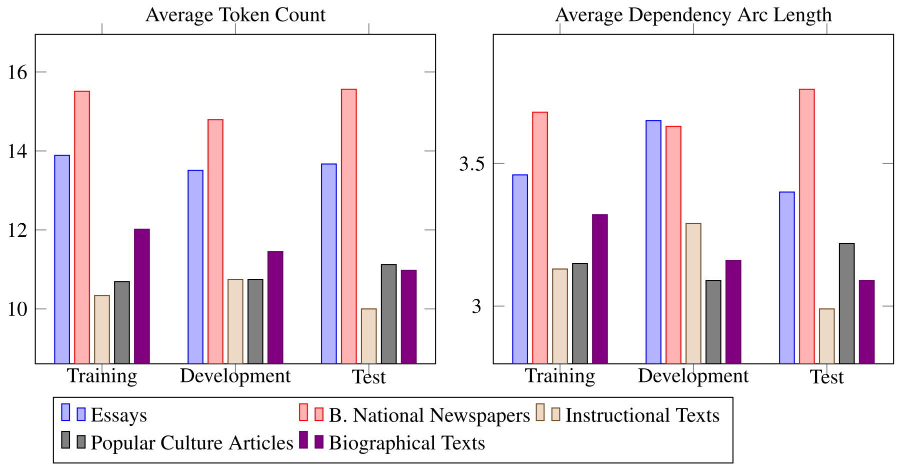


Fig. 2 The average token count and the average dependency arc length in a sentence for the five sections of the BOUN Treebank

Table 10 The performance of the parser on five different test sets according to UAS and LAS metrics

Training set	Training size	Test set	Test size	UAS F1-score	LAS F1-score
IMST-UD	3685	BOUN	979	69.38	58.65
BOUN	7803	BOUN	979	77.36	70.37
BOUN + IMST-UD	11,488	BOUN	979	77.57	70.50
IMST-UD	3685	IMST-UD	975	75.49	65.53
BOUN	7803	IMST-UD	975	73.63	62.92
BOUN + IMST-UD	11,488	IMST-UD	975	76.86	66.79
IMST-UD	3685	PUD	1000	65.28	49.50
BOUN	7803	PUD	1000	72.33	59.57
BOUN + IMST-UD	11,488	PUD	1000	72.76	60.39
IMST-UD	3685	BOUN + IMST-UD	1954	71.89	61.62
BOUN	7803	BOUN + IMST-UD	1954	75.67	66.99
BOUN + IMST-UD	11,488	BOUN + IMST-UD	1954	77.25	68.82
IMST-UD	3685	BOUN + IMST-UD + PUD	2954	69.03	56.37
BOUN	7803	BOUN + IMST-UD + PUD	2954	74.22	63.78
BOUN + IMST-UD	11,488	BOUN + IMST-UD + PUD	2954	75.30	65.17

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

fourth and fifth sets). Not surprisingly, the parser trained on IMST-UD performs better on its own test set (the second test set) than the parser model trained on the BOUN Treebank. However, the performance difference here is smaller than the one when these two models are tested on the BOUN Treebank's test set. To make a comparison, the parser trained on BOUN outperforms the parser trained on IMST-UD by 8% in UAS and by more than 10% in 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 parsing performance in terms of the 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 (Ballesteros et al., 2012; Foth et al., 2014).
- 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 were 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[18] as well as the 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 in sentences 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 types 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 main registers that the IMST-UD Treebank incorporates as indicated in the UD Project. This formality difference explains the lower usage of the `compound` relation type.

[18] The re-annotation process was performed on the UD 2.3 versions of these treebanks.

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

Relation type	IMST-UD (previous)	IMST-UD (re-annotated)	PUD (previous)	PUD (re-annotated)	BOUN
`acl`	1455 *(2.5%)*	1538 *(2.65%)*	–	515 *(3%)*	3494 *(2.85%)*
`acl:relcl`	–	–	514 *(3.04%)*	–	–
`advcl`	–	926 *(1.59%)*	405 *(2.4%)*	435 *(2.6%)*	2595 *(2.12%)*
`advcl:cond`	–	110 *(0.19%)*	–	13 *(0.07%)*	269 *(0.22%)*
`advmod`	1872 *(3.2%)*	2422 *(4.17%)*	1716 *(10.16%)*	1624 *(9.6%)*	5278 *(4.31%)*
`advmod:emph`	973 *(1.67%)*	976 *(1.68%)*	145 *(0.86%)*	143 *(0.8%)*	1724 *(1.41%)*
`amod`	3451 *(5.94%)*	3337 *(5.74%)*	1224 *(7.25%)*	1318 *(7.8%)*	7869 *(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`	2183 *(3.76%)*	2242 *(3.86%)*	694 *(4.1%)*	697 *(4.1%)*	3290 *(2.69%)*
`cc`	870 *(1.5%)*	879 *(3.1%)*	519 *(3.1%)*	520 *(3.1%)*	2800 *(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%)*	1512 *(1.24%)*
`clf`	–	8 *(0.01%)*	10 *(0.06%)*	10 *(0.06%)*	122 *(0.1%)*
`compound`	2219 *(3.82%)*	1977 *(3.40%)*	2012 *(11.91%)*	314 *(1.9%)*	2381 *(1.95%)*
`compound:lvc`	512 *(0.88%)*	522 *(0.90%)*	–	186 *(1.1%)*	1218 *(1.0%)*
`compound:redup`	199 *(0.34%)*	219 *(0.37%)*	–	9 *(0.05%)*	457 *(0.37%)*
`conj`	3718 *(6.40%)*	3529 *(6.07%)*	640 *(3.79%)*	696 *(4.1%)*	7250 *(5.92%)*
`cop`	813 *(1.40%)*	851 *(1.46%)*	517 *(3.06%)*	496 *(2.9%)*	1289 *(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`	2040 *(3.51%)*	1975 *(3.39%)*	671 *(3.97%)*	680 *(4%)*	4938 *(4.03%)*
`det:predet`	–	–	10 *(0.06%)*	8 *(0.05%)*	–

Table 11 continued

Relation type	IMST-UD (previous)	IMST-UD (re-annotated)	PUD (previous)	PUD (re-annotated)	BOUN
`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%)*	2039 *(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`	3780 *(6.51%)*	1870 *(3.22%)*	161 *(0.95%)*	174 *(1%)*	1371 *(1.12%)*
`nmod:arg`	–	–	110 *(0.65%)*	–	–
`nmod:poss`	3534 *(6.08%)*	3598 *(6.19%)*	722 *(4.27)*	1881 *(11%)*	10393 *(8.49%)*
`nsubj`	3747 *(6.45%)*	4430 *(7.63%)*	1023 *(6.05%)*	1238 *(7.3%)*	8499 *(6.94%)*
`nummod`	621 *(1.07%)*	567 *(0.98%)*	207 *(1.22%)*	263 *(1.6%)*	1568 *(1.28%)*
`obj`	4307 *(7.41%)*	3743 *(6.44%)*	816 *(4.83%)*	945 *(5.6%)*	7381 *(6.03%)*
`obl`	4444 *(7.65%)*	3824 *(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%)*	2150 *(12.72%)*	2148 *(12.7%)*	20,116 *(16.44%)*
`root`	5635 *(9.69%)*	5635 *(9.69%)*	1000 *(5.91%)*	1000 *(5.91%)*	9761 *(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%)*

Table 11 continued

Relation type	IMST-UD (previous)	IMST-UD (re-annotated)	PUD (previous)	PUD (re-annotated)	BOUN
Total	58,097	58,098	16,886	16,886	122,384

The normal numbers represent the counts and the italic numbers show their percentages

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 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.

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 converbs (Göksel & Kerslake, 2005) are considered nominal modifiers. However, these nominalized verbs actually construct embedded clauses and therefore are treated as clausal modifiers in the re-annotated treebank. In addition, the obl percentage decreased by 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.

7 Conclusion

In this paper, we presented the largest and the most comprehensive Turkish treebank with 9761 sentences: the BOUN Treebank. In the treebank, we encoded the surface forms of the sentences, the 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 also gave 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. Our treebank with a history of the changes we applied and our annotation guidelines are provided online.

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 reported UAS and LAS F1-scores with regards to specific text types and treebanks. We also showcased the results of the experiments where our new treebank was used with the re-annotated versions of the IMST-UD and PUD treebanks. All the tools and materials that are presented in this paper are available on our webpage https://tabilab.cmpe.boun.edu.tr/boun-pars.

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.

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

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

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

Declarations

Conflict of interest The authors declared that they have no conflict of interest.

Appendix 1: 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

Table 12 continued

<table>
<tr><th>Sak et al.</th><th>UD</th><th>Sak et al.</th><th>UD</th></tr>
<tr><td>Repeat</td><td>Mood=Iter</td><td>Inf</td><td>VerbForm=Vnoun</td></tr>
<tr><td>Hastily</td><td>Mood=Rapid</td><td>FutPart</td><td>VerbForm=Part| Tense=Future| Aspect=Imp</td></tr>
<tr><td>Almost</td><td>Mood=Pro</td><td>PastPart</td><td>VerbForm=Part| Tense=Past| Aspect=Perf</td></tr>
<tr><td>Stay</td><td>Mood=Dur</td><td>PresPart</td><td>VerbForm=Part| Tense=Pres</td></tr>
<tr><td>While</td><td>VerbForm=Conv| Mood=IOmp</td><td></td><td></td></tr>
</table>

Appendix 2: 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

Appendix 3: TNC registers

Table 15 TNC Details

ID	Section name	Number of words	%	Documents (total)	%
1	Academic prose: Medicine	71,446	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 etc.; excludes Hansard	59,445	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

Table 15 continued

ID	Section name	Number of words	%	Documents (total)	%
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	

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, 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 4th international workshop on Linguistically Interpreted Corpora (LINC-03)* at EACL 2003. https://www.aclweb.org/anthology/W03-2405.

Aygen, G. (2003). Extractability and the nominative case feature on tense. In S. Özsoy, D. Akar, M. Nakipoğlu-Demiralp, E. E. Taylan, & A. Aksu-Koç (Eds.), *Studies in Turkish linguistics: Proceedings of the 10th 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 M. S. Dryer & M. Haspelmath (Eds.), *The World atlas of language structures online*. Max Planck Institute for Evolutionary Anthropology.

Borges Völker, E., Wendt, M., Hennig, F., & Köhn, A. (2019). HDT-UD: A very large Universal Dependencies treebank for German. In *Proceedings of the third workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, Association for Computational Linguistics, Paris, France, pp. 46–57. https://doi.org/10.18653/v1/W19-8006, https://www.aclweb.org/anthology/W19-8006.

Brants, S., Dipper, S., Hansen, S., Lezius, W., & Smith, G. (2002). The Tiger treebank. In *Proceedings of the workshop on treebanks and linguistic theories* (Vol. 168).

Brants, T. (2000). TnT - a statistical part-of-speech tagger. *Sixth Applied Natural Language Processing Conference* (pp. 224–231). Association for Computational Linguistics. https://doi.org/10.3115/974147.974178.

Çetinoğlu, Ö., & Çöltekin, Ç. (2019). Challenges of annotating a code-switching treebank. In *Proceedings of the 18th international workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019)*, Paris, France, pp. 82–90, https://doi.org/10.18653/v1/W19-7809, https://www.aclweb.org/anthology/W19-7809

Çetinoğlu, Ö. (2009). A large scale LFG grammar for Turkish. Ph.D. 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, http://www.lrec-conf.org/proceedings/lrec2010/pdf/109_Paper.pdf

Çöltekin, Ç. (2015). A grammar-book treebank of Turkish. In M. Dickinson, E. Hinrichs, A. Patejuk, & A. Przepiórkowski (Eds.), *Proceedings of the 14th workshop on Treebanks and linguistic theories* (TLT 14) (pp. 35–49).

Çöltekin, Ç. (2016). (When) do we need inflectional groups? In *Proceedings of The 1st international conference on Turkic computational linguistics.*

Dozat, T., Qi, P., & Manning, C. D. (2017). Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *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. https://www.aclweb.org/anthology/W14-6108.

Eryiğit, G. (2007). ITU treebank annotation tool. In *Proceedings of the linguistic annotation workshop* (pp. 117–120).

Eryiğit, G., & Pamay, T. (2007). ITU validation set for Metu-Sabancı Turkish treebank. *Türkiye Bilis̀im 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. 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. (2001). The auxiliary ol at the morphology–syntax interface. In E. E. Taylan (Ed.), *The verb in Turkish*. John Benjamins.

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, 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. https://doi.org/10.18653/v1/W19-8010, 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. 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. 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. In *The free word order phenomenon: Its syntactic sources and diversity* (pp. 163–180). Mouton de Gruyter.

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, 28*, 498–519.

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

Makazhanov, A., Sultangazina, A., Makhambetov, O., & Yessenbayev, Z. (2015). Syntactic Annotation of Kazakh: Following the Universal Dependencies Guidelines. a report. In *3rd International conference on Turkic languages processing (TurkLang 2015)* (pp. 338–350).

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. 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. 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., 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. https://www.aclweb.org/anthology/L16-1262.

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., 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. http://www.lrec-conf.org/proceedings/lrec2006/pdf/223_pdf.pdf.

Nivre, J., Zeman, D., Ginter, F., & Tyers, F. M. (2017). Tutorial on Universal Dependencies. *Presented at European chapter of the Association for Computational Linguistics*, Valencia. Retrieved April 8, 2019, from http://universaldependencies.org/eacl17tutorial/applications.pdf.

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* (pp. 261–277). Springer. https://doi.org/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* (pp. 82–90). Tilburg University Press.

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

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

Öztürk, B. (2008). *Non-configurationality: Free word order and argument drop in Turkish*. The limits of syntactic variation (pp. 411–440). John Benjamins Publishing Company.

Öztürk, B. (2013). Postverbal constituents in SOV languages. In *Theoretical approaches to disharmonic word orders* (pp. 270–305). MIT.

Öztürk, B., & Taylan, E. E. (2016). Possessive constructions in Turkish. *Lingua, 182*, 88–108. https://doi.org/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, CO, USA, pp. 95–101. https://doi.org/10.3115/v1/W15-1610, 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. https://www.aclweb.org/anthology/C18-1324.

Sağ, Y. (2019). The semantics of number marking: Reference to kinds, counting, and optional classifiers. PhD thesis, Rutgers University.

Sak, H., Güngör, T., & Saraçlar, M. (2008). Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In *International conference on natural language processing* (pp. 417–427). Springer.

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*. Clarendon Press.

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* (Vol. 1: Long Papers), Association for Computational Linguistics, Sofia, Bulgaria, pp. 550–560. https://www.aclweb.org/anthology/P13-1054.

Sulubacak, U., & Eryiğit, G. (2018). 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. https://doi.org/10.2307/415636

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

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 Aras̀tırmaları Dergisi, 2*, 181–199.

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

Türk, U., Atmaca, F., Özateş, Ş. B., Öztürk Başaran, B., Güngör, T., & Özgür, A. (2019b). 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. https://doi.org/10.18653/v1/W19-8013, https://www.aclweb.org/anthology/W19-8013.

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.

van der Beek, L., Bouma, G., Malouf, R., & van Noord, G. (2002). The Alpino dependency treebank. In: Theune, M., Nijholt, A., & Hondorp, H. (Eds.), *12th Meeting on Computational Linguistics in the Netherlands (CLIN)* 2001, Rodopi, Language and Computers: Studies in Practical Linguistics, pp. 8–22, 30 November 2001.

Yıldız, O.T., Solak, E., Çandır, Ş., Ehsani, R., & Görgün, O. (2016). Constructing a Turkish constituency parse treebank. In *Information Sciences and Systems 2015* (pp. 339–347). Springer.

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. https://www.aclweb.org/anthology/W17-6532.

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. http://www.aclweb.org/anthology/K18-2001.

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. http://www.aclweb.org/anthology/K/K17/K17-3001.pdf.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.