A Multilayer Annotated Corpus for Turkish

Olcay Taner Yıldız, Koray Ak, Gökhan Ercan, Ozan Topsakal, Cengiz Asmazoğlu
Department of Computer Engineering, Işık University, İstanbul, Turkey
olcaytaner@isikun.edu.tr, {koray.ak, gokhan.ercan, ozan.topsakal, cengiz.asmazoglu}@isik.edu.tr

***Abstract*—In this paper, we present the first multilayer annotated corpus for Turkish, which is a low-resourced agglutinative language. Our dataset consists of 9,600 sentences translated from the Penn Treebank Corpus. Annotated layers contain syntactic and semantic information including morphological disambiguation of words, named entity annotation, shallow parse, sense annotation, and semantic role label annotation.**

***Index Terms*—Multilayer annotation, Turkish, named entity annotation, shallow parse annotation, semantic role labeling, word sense annotation, morphological annotation**

I. Introduction

In general, there are two central tasks in the field of natural language processing (NLP) studies. One of them is semantic analysis and the other one is syntactic analysis. Although it is perfectly possible to carry out a syntactic analysis of a sentence without understanding the meaning of any of the words, in order to fully comprehend a sentence, a computer has to understand not only the meanings of individual words, but also their hierarchical structure in the sentence. In the linguistics literature, a multilayer annotated corpus contains different syntactic and semantic layers for each sentence, thereby providing a great aid for NLP tasks.

Most of the NLP studies focus on analytic languages like English and many other Indo-European languages, whereas studies on agglutinative languages like Turkish are limited in this field. Agglutinative languages, in general, are arguably more difficult to work on than others, due to the fact that a word may get numerous different meanings via the use of morphological markers, such as affixes.

In this paper, we present the first multilayer annotated corpus for Turkish. The corpus currently contains 9,600 sentences, the original English counterparts of which are taken from Penn-Treebank. Annotated layers include morphological disambiguation of words, named entities, shallow parse, senses, and semantic role labels.

This paper is organised as follows: We define annotation layers in Section II and give the previous work in Section III. The details of our corpus and how it is constructed are given in Section IV. We provide the annotation statistics about the corpus in Section V and conclude in Section VI.

II. Annotation Layers

A. Morphological Disambiguation

In linguistics, the term morphology refers to the study of the internal structure of words. Each word is assumed to consist of one or more morphemes, which can be defined as the smallest linguistic unit having a particular meaning or grammatical function. One can come across morphologically simplex words, i.e. roots, as well as morphologically complex ones, such as compounds or affixed forms.

TABLE I
List of named entity types with the kinds of entities they belong to

Tag	Sample Categories
PERSON	people, characters
ORGANIZATION	companies, teams
LOCATION	regions, mountains, seas
TIME	time expressions
MONEY	monetarial expressions

Turkish is an agglutinative language, in which words are formed by attaching derivational and inflectional suffixes to the roots. Morphemes added to a word can change its part of speech, i.e., for instance, convert a noun to a verb - or vice versa -, or can create adverbs from adjectives. Moreover, during word formation, some letters can be changed or undergo deletion.

Morphological disambiguation is the process of identifying the correct morphological analysis of a word. For example, the Turkish noun "sorunu" has three morphological analyses, as shown below:

sorun + NOUN + A3SG + PNON + ACC (the problem)
sorun + NOUN + A3SG + P3SG + NOM (her/his problem)
soru + NOUN + A3SG + P2SG + ACC (your question)

Depending on the context, i.e. based on its intended meaning, one needs first to identify the root word and then choose the correct morphological analysis.

B. Named Entity Tagging

Anything that is denoted by a proper name, i. e., for instance, a person, a location, or an organization, is considered to be a named entity. In addition, named entities also include things like dates, times, or money. Here is a sample text with named entities marked (See Table I for typical generic named entity types).

[$_{ORG}$ Türk Hava Yolları] bu [$_{TIME}$ Pazartesi'den] itibaren [$_{LOC}$ İstanbul] [$_{LOC}$ Ankara] hattı için indirimli satışlarını [$_{MONEY}$ 90 TL'den] başlatacağını açıkladı.
[$_{ORG}$Turkish Airlines] announced that from this [$_{TIME}$ Monday] on it will start its discounted fares of [$_{MONEY}$ 90TL] for [$_{LOC}$ İstanbul] [$_{LOC}$ Ankara] route.

In named entity recognition (NER), one tries to find the strings within a text that correspond to proper names (exclud-

TABLE II
LIST OF SHALLOW PARSE CHUNK TAGS

Tag	Question
ÖZNE	Who, What
ZARF TÜMLECİ	When, How, Why
DOLAYLI TÜMLEÇ	Where, To/From whom
NESNE	What, Whom
YÜKLEM	Predicate

ing TIME and MONEY) and classify the type of entity denoted by these strings. The standard approach for NER is a word-by-word classification, where the classifier is trained to label the words in the text with tags that indicate the presence of particular kinds of named entities. After giving the class labels (named entity tags) to the data, the next step is to select a group of features to distinguish between different named entities for each input word.

The NER problem is difficult partly due to the ambiguity in sentence segmentation; one needs to extract which words belong to a named entity, and which not. Another difficulty occurs when some word may be used as a name of either a person, an organization or a location. For example, *Deniz* may be used as the name of a person, or - within a compound - it can refer to a location *Marmara Denizi* "Marmara Sea", or an organization *Deniz Taşımacılık* "Deniz Transportation".

C. Shallow Parsing

Many language processing tasks do not require complex parse trees. Instead, a partial parse, or a shallow parse of a sentence is sufficient. Shallow parsing is the process of identifying the flat, non-overlapping parts of a sentence. These parts typically include Özne (Subject), Yüklem (Predicate), Nesne (Object), and Tümleç (Adverbial Clause), which is further divided into Zarf Tümleci and Dolaylı Tümleç in Turkish. Since a parsed text does not include a hierarchical structure, a bracketing notation is sufficient to denote the location and the type of shallow parse chunks in a sentence. Here is a sample text with shallow parse chunks marked (See Table II shows typical shallow parse tags and the questions asked to the predicate to identify the chunks for those tags).

[$_{OZNE}$ Türk Hava Yolları] [$_{ZARF\ TUMLECI}$ Salı günü] [$_{NESNE}$ yeni indirimli fiyatlarını] [$_{YUKLEM}$ açıkladı]
[$_{SUBJECT}$ Turkish Airlines] [$_{PREDICATE}$ announced] [$_{OBJECT}$ new discounted fares] [$_{ADVERBIAL\ CLAUSE}$ on Tuesday]

In shallow parsing, one tries to find the strings of text that belong to a chunk and to classify the type of that chunk. The standard approach for shallow parsing is a word-by-word classification, where the classifier is trained to label the words in the text with tags that indicate the presence of particular chunks. After giving class labels to the data, the next step is to select a group of features to discriminate among the different chunks for each input word.

D. Word Sense Disambiguation

The task of choosing the correct sense for a word is called word sense disambiguation (WSD). WSD algorithms take an input word w within a context, which has a fixed set of potential word senses S_w, and produce an output chosen from S_w. In the isolated WSD task, one usually uses the set of senses from a dictionary or theasurus like WordNet. Table III shows an example for the word 'yüz', which can refer to the number '100', to the verb 'swim' or to the noun 'face'.

TABLE III
POSSIBLE DEFINITIONS FOR THE SENSE TAGS FOR YÜZ

Sense	Definition
yüz^1 (hundred)	The number coming after ninety nine
yüz^2 (swim)	move or float in water
yüz^3 (face)	face, visage, countenance

In the literature, there are actually two variants of the generic WSD task. In a **lexical sample** task, a small selected set of target words is chosen, along with a set of senses for each target word. For each target word w, a number of corpus sentences (context sentences) are manually labeled with the correct sense of w. In an **all-words** task, systems are given entire sentences and a lexicon with the set of senses for each word in each sentence. Annotators are then asked to disambiguate every word in the text.

In all-words WSD, a classifier is trained to label the words in the text with their set of potential word senses. After giving the sense labels to the words in the data, the next step is to select a group of features to distinguish the different senses for each input word.

E. Semantic Role Labeling

Semantic Role Labeling (SRL) is a well-defined task where the objective is to analyze propositions expressed by the verb. In SRL, each word that bears a semantic role in the sentence has to be identified. There are different types of arguments (also called 'thematic roles') such as Agent, Patient, Instrument, and also of adjuncts, such as Locative, Temporal, Manner, and Cause. These arguments and adjuncts represent entities participating in the event and give information about the event characteristics.

In the field of SRL, PropBank [1] is one of the studies widely recognized by the computational linguistics communities. PropBank is the bank of propositions where predicate-argument information of the corpora is annotated, and the semantic roles or arguments that each verb can take are posited.

Each verb has a frame file, which contains arguments applicable to that verb. Frame files may include more than one roleset with respect to the senses of the given verb. In the roleset of a verb sense, argument labels Arg0 to Arg5 are described according to the meaning of the verb. For the example below, the predicate is "announce" from PropBank, Arg0 is "announcer", Arg1 is "entity announced", and ArgM-TMP is "time attribute".

[$_{ARG0}$ Türk Hava Yolları] [$_{ARG1}$ indirimli satışlarını] [$_{ARGM-TMP}$ bu Pazartesi] [$_{PREDICATE}$ açıkladı].

TABLE IV
LIST OF NAMED SEMANTIC ROLE TYPES

Tag	Meaning	Tag	Meaning
Arg0	Agent or Causer	ArgM-EXT	Extent
Arg1	Patient or Theme	ArgM-LOC	Locatives
Arg2	Instrument, start point, end point, beneficiary, or attribute		
ArgM-CAU	Cause	ArgM-MNR	Manner
ArgM-DIS	Discourse	ArgM-ADV	Adverbials
ArgM-DIR	Directionals	ArgM-PNC	Purpose
ArgM-TMP	Temporals		

[$_{ARG0}$ Turkish Airlines] [$_{PREDICATE}$ announced] [$_{ARG1}$ its discounted fares] [$_{ARGM-TMP}$ this Monday].

Table IV shows typical semantic role types. Only Arg0 and Arg1 indicate the same thematic roles across different verbs: Arg0 stands for the Agent or Causer and Arg1 is the Patient or Theme. The rest of the thematic roles can vary across different verbs. They can stand for Instrument, Start point, End point, Beneficiary, or Attribute. Moreover, PropBank uses ArgM's as modifier labels indicating time, location, temporal, goal, cause etc., where the role is not specific to a single verb group; it generalizes over the entire corpus instead.

III. Previous Work

A. NER

Turkish-specific NER data are relatively scarce when compared to languages that have a wider global distribution. First work on Turkish NER [2] presents a study based on information extraction on data, gathered from news reports. An experiment-based study on tweets [3] shows the difference between processing a formal syntax over a social media text. Yet in tweets there is no need to follow spelling rules, and words and even letters in the case of emoticons and some informal abbreviations can be used in different senses than usual.

B. Shallow Parsing

In another study, Yildiz et. al. [4] manually extract data from Penn Treebank. Upon translating the data into Turkish, they try to automatically identify and tag the chunks.

Kutlu & Cicekli [5] work on noun phrase chunking in Turkish. They use hand-crafted rules for the dependency parser that are suitable for complex sentences due to their flexibility.

Finally, El-Kahlout & Akin [6] propose two different techniques. In the former, chunks are extracted according to the results of the Turkish dependency parser. In the latter, annotated Turkish sentences are used by a CRF-based chunker, which is enhanced with morphological and combinatorial features.

C. Word Sense Disambiguation

Ilgen et al. [7] aim to find out the best sets of collocational features for WSD in Turkish language. They use a lexical dataset which includes polysemous nouns and verbs.

Altintas et al. [8] look into the effects of windowing on success rates in WSD.

Ilgen et al. [9] investigate the effects of different windowing schemes on word sense disambiguation accuracy in Turkish language. They use a Turkish lexical sample dataset in their experiments.

In their paper, Orhan and Altan [10] investigate feature selection strategies for word sense disambiguation task in Turkish. The paper explains that Turkish verbs can be affected by many different factors on the sense disambiguation process.

D. Semantic Role Labeling

Recently, Şahin presented their study [11], [12] for Turkish PropBank frames generation using Crowdsourcing techniques. Verb sense annotation prior to the frame creation task is achieved by using Crowd intelligence. In the construction phase ITU-METU-Sabancı Treebank (IMST) is used as a resource. Frame files for 1,262 verb senses are generated in the study.

IV. Corpus

The original data for our corpus is drawn from the Penn-Treebank corpus. Selected sentences from this Penn-Treebank corpus containing less than 15 words are translated into Turkish [13]. The corpus currently contains 9,600 sentences.

A. Annotation Setup

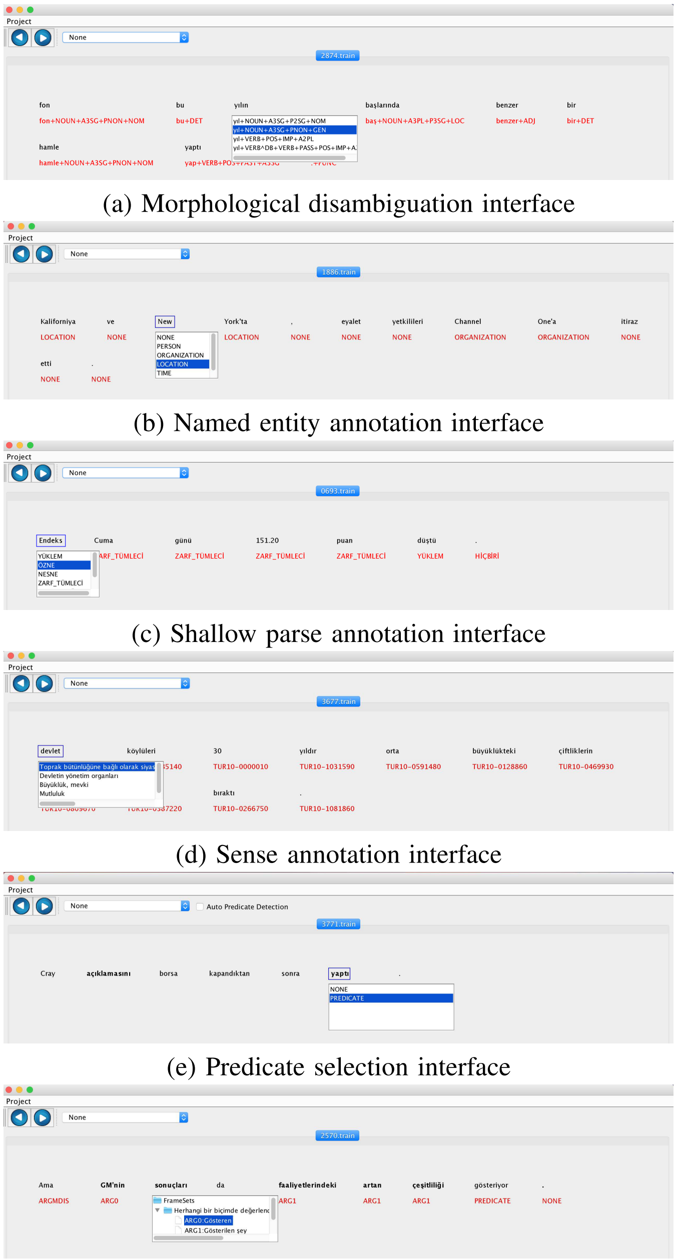
For the annotation, we are using an in-house NLP Toolkit, which supports all the operations mentioned in the sections above. To accomplish the annotation, we integrated corresponding editors (morphological disambiguation editor, entity annotation editor, shallow parse editor, word sense annotation editor, predicate editor, argument editor) to our toolkit in order to use the same infrastructure.

The same annotation logic applies for all editors. Words in the middle area are clickable. Once the user clicks a word, a possible item (morphological analyses for morphological disambiguation, entity tags for entity annotation, shallow parse tags for shallow parsing, distinct senses for word sense annotation, roles for argument annotation) that can be assigned to the node pops up. After the selection is made, the selected item is printed below the node. Since we use translated sentences from Penn Treebank, all sentences have their English counterparts. These sentences can guide and help annotators to check their annotation.

We worked with six annotators, all undergraduate students of Işık University. Video guidelines for all editors for annotation were prepared based on guidelines provided by linguists. These annotators were trained before starting to annotate files. The corpus was divided into six equal parts and each part was assigned to a single student.

B. Interface

1) Morphological Disambiguation: In the morphological disambiguation, human annotators select the correct morphological analysis among all possible ones returned from the automatic parser [14] (See Figure IV-B1(a)). The tag set and morphological representation were adopted from the same study. Each output of the parser comprises the root of the word, its part-of-speech tag and a set of morphemes, each separated



(a) Morphological disambiguation interface

(b) Named entity annotation interface

(c) Shallow parse annotation interface

(d) Sense annotation interface

(e) Predicate selection interface

(f) Semantic role labeling interface

with a '+' sign. For the out of vocabulary words, the morphological parser gives the output "word+NOUN+A3SG+NOM".

2) Entity Annotation: In the entity annotation, the annotators annotate named entities in a sentence (See Figure IV-B1(b)). Possible named entities are listed in Table I. If a word is not a named entity (a regular word, a punctuation, etc.), the user selects the tag "NONE".

3) Shallow Parsing: In the shallow parsing step, the annotators choose the correct shallow parse tag for each word in a sentence (See Figure IV-B1(c)). Possible shallow parse tags are listed in Table II. If a word does not fall into one of the established categories of parse tags, the user selects the tag "HİÇBİRİ". If there are multiple subsentences connected via conjunctions, such as 've' (and) or 'veya' (or), the user analyses these subsentences independently.

4) Word Sense Disambiguation: For the word sense disambiguation task, the annotators choose the correct sense for each word (including punctuation marks) within a sentence (See Figure IV-B1(d)). All possible senses for a root word (taken from TDK dictionary) are listed in the droplist.

TDK dictionary is Turkish Language Institution's (a governmental organization, abbreviated as TDK) dictionary, which is a collection of 92,371 distinct lemmas organized in 121,602 sense entries. We stored the TDK dictionary in XML format. In our format, units that constitute the vocabulary are possible meanings of the words. We named these units 'synsets', as conventional in the domain of wordnets, but our synsets are not merged or interlinked with other synsets. In fact, we have not made extra processing on the original dictionary; instead, we transfigured it into an XML schemata. The structure of a sample synset is as follows:

```
<SYNSET>
<ID>TUR10-0038510</ID>
<LITERAL>anne
<SENSE>2</SENSE>
</LITERAL>
<POS>n</POS>
<DEF>...</DEF>
<EXAMPLE>...</EXAMPLE>
</SYNSET>
```

Each entry in the dictionary is enclosed by <SYNSET> and </SYNSET> tags. Synset members are represented as literals and their sense numbers. <ID> shows the unique identifier given to the synset. <POS> and <DEF> tags denote part of speech and definition, respectively. As for the <EXAMPLE> tag, it gives a sample sentence for the synset.

Another issue that must be handled by the sense disambiguation tool is collocations. Many English words have a multi-word translation into Turkish and they need special attention to obtain a sense list. As a solution, we take cartesian product of derived forms of each word and search the dictionary for each combination. If any senses are found, we add them into the sense lists of the words that are included in the collocation. For instance, consider the following parallel sentences:

New York'ta kendine geldi
He came to himself in New York

In this sentence, there are two collocations, namely "New York" and "kendine gelmek". They correspond to "New York" and "come to oneself" in the English side. The available senses displayed in the droplist for the word "geldi" contain both the possible senses of the simplex "gelmek", and the ones returned for the multi-word expression "kendine gelmek". Similarly, the displayed senses for the word "kendine" are composed of the senses of the simplex "kendi", as well as the ones returned for "kendine gelmek".

5) Verbal Predicate Selection: In the verbal predicate selection step, the annotators choose the verbal predicate in each sentence (See Figure IV-B1(e)). If the predicate is a multi-word expression, the user needs to select all constituents of this expression with the tag "PREDICATE". For example, the verbal predicate of the sentence "Köy hayatı ona iyi geldi" (Village life came to him well) is "iyi geldi" (came well) and the annotator should tag both words as "PREDICATE".

Similar to the shallow parsing step, if there are multiple subsentences in a sentence, verbal predicates of all those subsentences are annotated "PREDICATE". For example, there are two verbal predicates in the sentence "Aysu topu attı ve Kerem onu yakaladı" (Aysu threw the ball, and Kerem catched it), namely "attı" (threw) and "yakaladı" (catched). The annotator should tag both words as "PREDICATE".

If there is no verbal predicate in the sentence, the annotator leaves the sentence unmarked.

6) Semantic Role Labeling: Given the verbal predicate(s) in a sentence, as a last step, the annotators annotate semantic roles of words (See Figure IV-B1(f)). The list of semantic roles are determined with respect to the frameset of the selected verbal predicate for that sentence.

Unlike the original PropBank frame files, where each verb has a file with different rolesets for each different sense, we decided to use a single xml file, which contains all verbs and their respective senses for the sake of simplicity in the current architecture. The structure of a sample frameset is as follows [15]:

```
<FRAMESET id="0006410">
<ARG name="ARG0">Açan</ARG>
<ARG name="ARG1">Açılan şey</ARG>
<ARG name="ARGMTMP">Açılma zamanı</ARG>
</FRAMESET>
```

Each entry in the frame file is enclosed by <FRAMESET> and </FRAMESET> tags. *id* shows the unique identifier given to the frameset, which is the same ID in the synset file of the corresponding verb sense. <ARG> tags denote the semantic roles of the corresponding frame.

If there are multiple verbal predicates in the sentence, the framesets of these predicates are shown separately. For example, in the sentence above "Aysu topu attı ve Kerem onu yakaladı" (Aysu threw the ball, and Kerem catched it); for the word "topu" (the ball), the annotation tool shows the semantic roles of both "atmak" (throw) and "yakalamak" (catch).

C. Data Format

The words in the original sentence are separated via spaces. After all six steps of processing are completed, the data structure stored for the same word has the following form in our system:

```
{turkish=yatırımcılar}
{analysis=yatırımcı+NOUN+A3PL+PNON+NOM}
{semantics=0841060}{namedEntity=NONE}
{shallowParse=ÖZNE}{propbank=ARG0:0006410}
```

TABLE V
INTER-ANNOTATOR AGREEMENT

Layer	Agreement	Expected Agreement	Cohen's Kappa
NER	0.975	0.167	0.969
Shallow Parse	0.79	0.167	0.748
Sense Annotation	0.785	0.545	0.527

TABLE VI
10 MOST-FREQUENT SURFACEFORMS EXCEPT STOP-WORDS

Surfaceform	Count	Surfaceform	Count
bay (mr.)	481	büyük (large)	234
olarak (as)	351	milyar (billion)	224
milyon (million)	327	ediyor (do, accept)	177
amerikan (American)	252	oldu (did)	174
hisse (share, stock)	235	şirket (company, firm)	173

As is self-explanatory, "turkish" tag shows the original Turkish word; "analysis" tag shows the correct morphological parse of that word; "semantics" tag shows the ID of the correct sense of that word; "namedEntity" tag shows the named entity tag of that word; "shallowParse" tag shows the semantic role of that word; "propbank" tag shows the semantic role of that word for the verb synset id (frame id in the frame file) which is also given in that tag. Annotated corpus and source codes are freely available[1].

V. RESULTS

A. Inter-annotator Agreement

For the evaluation of the annotated dataset, we used an inter-annotator agreement measure. Two different group of annotators annotated the same sentences. Due to a lack of time, we could only re-annotate 100 of the total number of 9,600 sentences. Inter-annotator agreement scores, expected agreement scores and Cohen's kappa coefficients are given in Table V. The expected inter-annotator agreement is calculated by assuming that the annotators annotated completely randomly.

B. Statistics

In the corpus, word frequencies and the coverage of senses are not balanced. The result of the current annotation effort is a corpus of about 88,359 word occurrences. There are 20,637 distinct surfaceforms (including punctuation and stop-words) and 177 of them occur 50 or more times in the corpus. The average number of samples per lemma is equal to 4.28. Table VI lists 10 most-frequent surfaceforms except stop-words and pronouns.

After correctly morphological disambiguation of 81,580 words, there are 9,328 distinct root words (including punctuation and stop-words) and 240 of them occur 50 or more times in the corpus. The average number of samples per root word is equal to 8.75. Table VII lists 10 most-frequent root words except stop-words and pronouns. Table VIII lists the frequencies of POS tags of the root words.

Given the named entity tag categories, the distribution of the NER data is shown in Table IX. As expected, most of the

[1] http://haydut.isikun.edu.tr/nlptoolkit.html

TABLE VII
10 MOST-FREQUENT ROOT WORDS EXCEPT STOP-WORDS

Root word	Count	Root word	Count
olmak (be)	1421	hisse (share, stock)	380
etmek (do)	796	dolar (dollar)	373
bay (mr.)	476	milyon (million)	362
yapmak (do)	391	artmak (increase)	347
şirket (company, firm)	385	gelmek (come)	326

TABLE VIII
DISTRIBUTION OF POS TAGS OF THE ROOT WORDS

Word Type	Count	Word Type	Count
Noun	34,433	Number	4,240
Punctuation	13,896	Adverb	2,994
Verb	11,964	Pronoun	1,049
Adjective	6,643		

TABLE IX
DISTRIBUTION OF THE NAMED ENTITY TAGS

Tag	Count	Percentage
ORGANIZATION	4,418	5.05
PERSON	2,612	2.99
MONEY	2,240	2.56
LOCATION	1,303	1.49
TIME	1,194	1.37
NONE	75,682	86.54
Total	87,449	100.00

TABLE X
DISTRIBUTION OF THE SHALLOW PARSE TAGS

Tag	Count	Percentage
NESNE	15,768	20.02
ÖZNE	13,996	17.77
ZARF TÜMLECİ	13,003	16.51
YÜKLEM	11,607	14.74
DOLAYLI TÜMLEÇ	7,252	9.21
HİÇBİRİ	17,134	21.75
Total	78,760	100.00

TABLE XI
DISTRIBUTION OF THE SEMANTIC ROLES

Tag	Count	Tag	Count
ARG0	716	ARGM-LOC	74
ARG1	1,066	ARGM-EXT	67
ARG2	55	ARGM-DIS	41
ARGM-MNR	157	ARGM-ADV	24
ARGM-TMP	103	ARGM-CAU	13

words are not named entities (over 86 percent of the words are annotated with “NONE”).

Regarding shallow parse, the distribution of the chunk types is shown in Table X. If the sentence does not contain a predicate, and hence is considered to be corrupted, all of the words are annotated as “HİÇBİRİ”. Similarly, if there are words which do not constitute any direct relation to the predicate, they are also annotated as “HİÇBİRİ”. For that reason, approximately one quarter of all words are annotated with tag “HİÇBİRİ”.

Given the semantic role label categories, the distribution of the semantic roles is shown in Table XI.

VI. CONCLUSION

In this paper, we introduced a multi-layered (i.e. syntactic and semantic layers) annotated corpus for Turkish, a low-resourced agglutinative language. Although not all the layers are fully annotated yet, the corpus currently consists of over 9,600 sentences. The preliminary version of this dataset was previously used in NER [16], shallow parsing [17], and WSD [18] tasks.

ACKNOWLEDGEMENTS

This work was supported by Tübitak project 116E104 and Işık University BAP project 15B201.

REFERENCES

[1] M. Palmer, D. Gildea, and P. Kingsbury, “The proposition bank: An annotated corpus of semantic roles,” *Comput. Linguist.*, vol. 31, no. 1, pp. 71–106, Mar. 2005. [Online]. Available: http://dx.doi.org/10.1162/0891201053630264

[2] G. Tür, D. Hakkani-Tür, and K. Oflazer, “A statistical information extraction system for Turkish,” *Natural Language Engineering*, vol. 9, pp. 181–210, 2003.

[3] D. Küçük and R. Steinberger, “Experiments to improve named entity recognition on Turkish tweets,” *arXiv:1410.8668*, 2014.

[4] O. T. Yildiz, E. Solak, R. Ehsani, and O. Görgün, “Chunking in Turkish with conditional random fields,” in *CICLing*, 2015, pp. 173–184.

[5] M. Kutlu and I. Cicekli, “Noun phrase chunking for Turkish using a dependency parser,” in *ISCIS*, 2015, pp. 381–391.

[6] I. D. El-Kahlout and A. A. Akin, “Turkish constituent chunking with morphological and contextual features,” in *CICLing*, 2013, pp. 270–281.

[7] B. İlgen, E. Adalı, and A. C. Tantuğ, “The impact of collocational features in Turkish word sense disambiguation,” in *16th International Conference on Intelligent Engineering Systems*, 2012.

[8] E. Altıntaş, E. Karslıgil, and V. Coşkun, “The effect of windowing in word sense disambiguation,” in *Computer and Information Sciences*, vol. 3733, 2005.

[9] B. İlgen, E. Adalı, and A. Tantuğ, “A comparative study to determine the effective window size of Turkish word sense disambiguation systems,” in *Information Sciences and Systems*, vol. 264, 2013.

[10] Z. Orhan and Z. Altan, “Impact of feature selection for corpus-based wsd in Turkish,” in *MICAI 2006: Advances in Artificial Intelligence*, vol. 4293, 2006.

[11] G. G. Şahin, “Framing of verbs for turkish propbank,” in *TurCLing 2016 in conj. with 17th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2016)*, 2016.

[12] ——, “Verb sense annotation for turkish propbank via crowdsourcing,” in *17th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2016)*, 2016.

[13] O. T. Yildiz, E. Solak, O. Gorgun, and R. Ehsani, “Constructing a Turkish-English parallel treebank,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 112–117.

[14] O. Gorgun and O. T. Yildiz, “A novel approach to morphological disambiguation for Turkish,” in *International Conference on Computer and Information Science*, 2011, pp. 77–83.

[15] K. Ak, C. Toprak, V. Esgel, and O. T. Yildiz, “Construction of Turkish proposition bank,” *Turkish Journal of Electrical Engineering & Computer Sciences*.

[16] B. Ertopcu, A. B. Kanburoglu, O. Topsakal, O. Acikgoz, A. T. Gurkan, B. Ozenc, I. Cam, B. Avar, G. Ercan, and O. T. Yildiz, “A new approach for named entity recognition,” in *International Conference on Computer Science and Engineering*, 2017, pp. 474–479.

[17] O. Topsakal, O. Acikgoz, A. T. Gurkan, A. B. Kanburoglu, B. Ertopcu, B. Ozenc, I. Cam, B. Avar, G. Ercan, and O. T. Yildiz, “Shallow parsing in Turkish,” in *International Conference on Computer Science and Engineering*, 2017, pp. 480–485.

[18] O. Acikgoz, A. T. Gurkan, B. Ertopcu, O. Topsakal, B. Ozenc, A. B. Kanburoglu, I. Cam, B. Avar, G. Ercan, and O. T. Yildiz, “All-words word sense disambiguation for Turkish,” in *International Conference on Computer Science and Engineering*, 2017, pp. 490–495.