Constructing a Turkish Constituency Parse TreeBank

Olcay Taner Yıldız, Ercan Solak, Şemsinur Çandır, Razieh Ehsani and Onur Görgün

Abstract In this paper, we describe our initial efforts for creating a Turkish constituency parse treebank by utilizing the English Penn Treebank. We employ a semi-automated approach for annotation. In our previous work [18], the English parse trees were manually translated to Turkish. In this paper, the words are semi-automatically annotated morphologically. As a second step, a rule-based approach is used for refining the parse trees based on the morphological analyses of the words. We generated Turkish phrase structure trees for 5143 sentences from Penn Treebank that contain fewer than 15 tokens. The annotated corpus can be used in statistical natural language processing studies for developing tools such as constituency parsers and statistical machine translation systems for Turkish.

1 Introduction

Treebanks annotated for the syntactic or semantic structures of the sentences are essential for developing state-of-the-art statistical natural language processing (NLP) systems including part-of-speech-taggers, syntactic parsers, and machine translation systems. There are two main groups of syntactic treebanks, namely treebanks annotated for constituency (phrase structure) and the ones that are annotated for dependency structure. The first large-scale treebank, the Penn Treebank, was developed for English and annotated for constituency structures of sentences [13]. The development of other treebanks followed for a wide variety of languages including German [3], French [1], Arabic [12], Chinese [17], Hungarian [7], and Finnish [10].

In this study, we report our preliminary efforts for constructing a Turkish constituency parse treebank corpus. Turkish is morphologically rich language with a highly agglutinative nature. Sentences in general have an SOV order. However,

O.T. Yıldız (✉) · E. Solak · Ş. Çandır · R. Ehsani · O. Görgün
Işık University, Istanbul, Turkey
e-mail: olcaytaner@isikun.edu.tr

O. Görgün
Alcatel Lucent Teletaş Telekomünikasyon A.Ş., Istanbul, Turkey

O.H. Abdelrahman et al. (eds.), *Information Sciences and Systems 2015*,
Lecture Notes in Electrical Engineering 363,
DOI 10.1007/978-3-319-22635-4_31

constituents can also be reordered for emphasizing and topicalizing certain elements based on the discourse. In addition, case markings (e.g. locative, dative) play crucial roles in identify the syntactic functions of the constituents [11].

We utilize a subset of the English Penn Treebank and introduce a semi-automatic annotation approach consisting of three phases. In our previous work [18], the parse trees in Penn Treebank were manually translated to Turkish. In this paper, the words are automatically annotated morphologically, and refined manually. As a second step, a rule-based method is developed to refine the parse trees based on the morphological annotations of the words. To the best of our knowledge, this is the first Turkish treebank annotated for phrase structure. We believe such a resource will promote statistical NLP research and applications for Turkish.

The paper is organized as follows: In Sect. 2, we give the literature review for treebank construction efforts in Turkish. We give the details of our corpus construction strategy in Sect. 3. Finally, we conclude in Sect. 4.

2 Related Work

There are only a handful of studies for creating Turkish treebank corpora. The METU-Sabancı Dependency Treebank is the first Turkish treebank [2]. It contains 7262 sentences manually annotated morphologically and syntactically. The syntactic annotation consists of head-dependent relations and functional categories. The METU-Sabancı Treebank has been used in several Turkish NLP studies [8, 9, 15, 16, 19].

There have also been some efforts for transforming the dependency-structure representation of METU-Sabancı Treebank into different syntactic representations. Cakici [4] extracted a Combinatory Categorical Grammar (CCG) from the METU-Sabancı Treebank with annotation of lexical categories. A finite state machine based approach was developed for generating a Lexical Grammar Formalism (LGF) for Turkish by using the sub-lexical units that reveal the internal structures of the words in [5, 6].

To our knowledge, this study is the first effort for creating a Turkish treebank corpus annotated for phrase-structures. The sentences in the corpus are selected from the English Penn Treebank. Therefore, besides other statistical NLP studies, the corpus can also be used for English-Turkish machine translation studies as a parallel treebank.

3 Corpus Construction Strategy

In this initial phase of our annotation efforts, we selected sentences that have at most 15 tokens, including punctuations, from the Penn Treebank II [13]. This choice reduced the total number of trees in the Penn Treebank to 9560 including 8660 trees

from the training set, 540 trees from the test set and 360 trees from the development set. We translated all of the test and development sets and nearly half of trees in the training set and obtained a total of 5143 Turkish trees.

Our corpus construction strategy is semi-automatic consisting of three stages. In the first stage, we manually translated the selected trees from the Penn Treebank. The last two stages are mostly automatic. In the second stage, we first automatically annotated words morphologically and then human annotators manually selected the correct morphological annotation for words having more than one possible morphological analyses. In the final stage, we refined the constituency parse trees by using the morphological analyses from stage two and following a set manually designed rules. In this paper, we emphasize on the last two steps, but for the sake of completeness, we also explain the first stage, which was completed in our previous paper [18].

3.1 Translation of Trees

We built a tool to facilitate translators' task. The tool both visualizes trees and makes the tree manipulation task much easier and faster. Additionally, the tool recommends glosses to translators based on the statistics of previously translated trees. Hence, as the number of translated trees increases, the translation task gets easier.

We constrain the translations of trees to two operations. We can only permute the children of a node and replace the leaf nodes with translated glosses. Since Turkish is an agglutinative language, it is often the case that we embed an English constituent in the morphemes of a Turkish stem. In such cases, we replace the English constituent leaf with *NONE*.

Turkish sentences generally have the SOV order. When translating English trees, subtrees are permuted to follow that order. Additionally, Turkish morphotactics determine the order of constituents.

We obtain the translated tree in Fig. 1 after following the above rules. Note that (VB talk), (RB not), (MD should), and (PRP you) are embedded in the morphological analysis "konuş-NEG-NECES-2SG" of the verb "konuşmamalısın".

3.2 Morphological Analysis

In general, functional words in English correspond to specific morphemes attached to the word stem in Turkish. For instance, "I/PRP will/MD not/RB do/VB" is translated to Turkish as "yap-ma-yacağ-ım" which corresponds to "do-NEG-FUT-1SG" in English. Hence, even though annotating trees syntactically is sufficient for English, morphological analysis is necessary for Turkish because of its highly agglutinative structure.

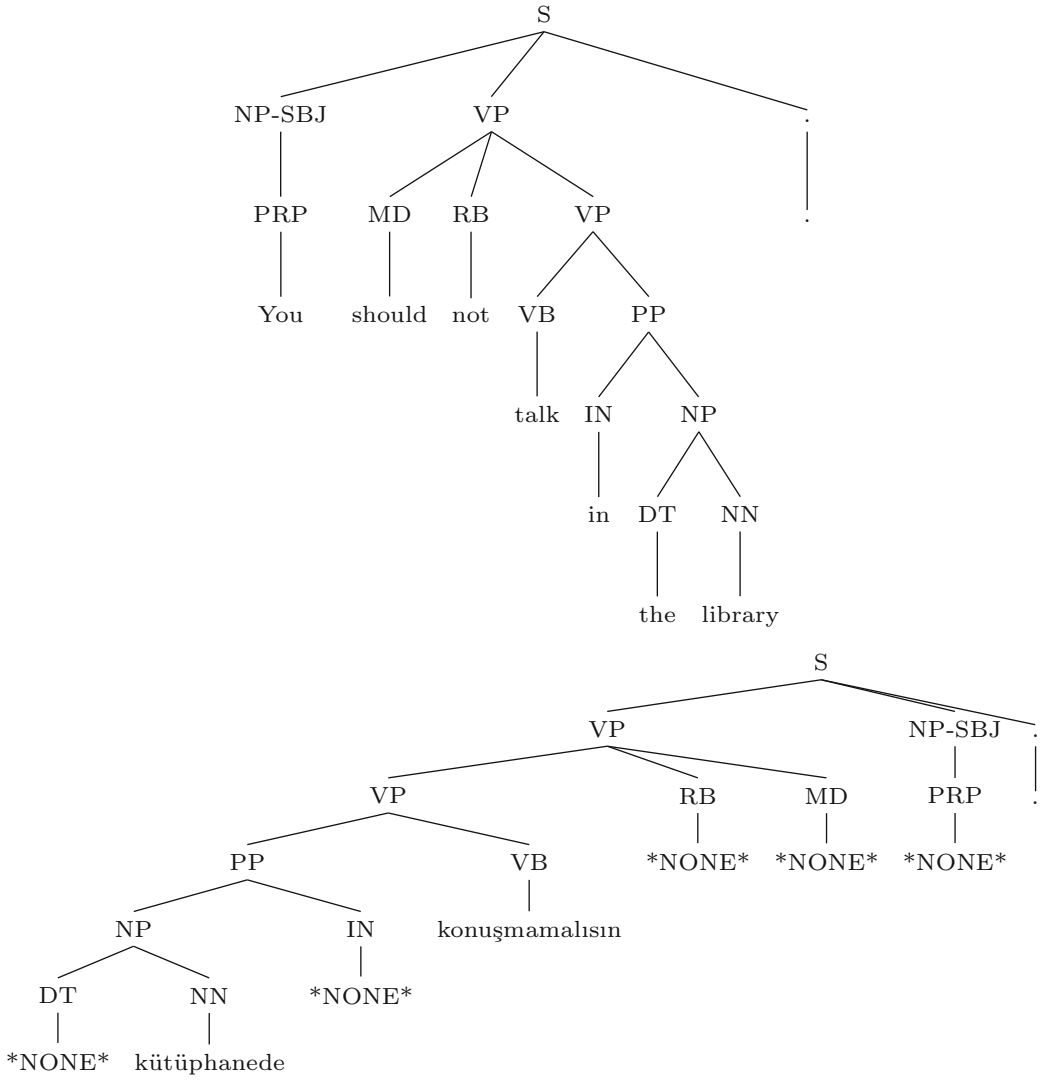


Fig. 1 Original and translated trees, kütüphane-de konuş-ma-malı-sın (library-LOC talk-NEG-NECES-2SG)

This stage consists of two substages. First, words are morphologically analyzed using a analyzer implementation based on Oflazer's [14]. Second, human annotators manually select the correct analysis using a graphical user interface. Figure 2 shows the morphological analysis of the translated tree in Fig. 1.

3.3 Morphological Annotation

This final stage gets translated trees with morphological analyses as input. We perform morphological annotation to translated trees, in addition to the syntactic annotation. We follow the set of rules described below to obtain the final Turkish constituency parse trees.

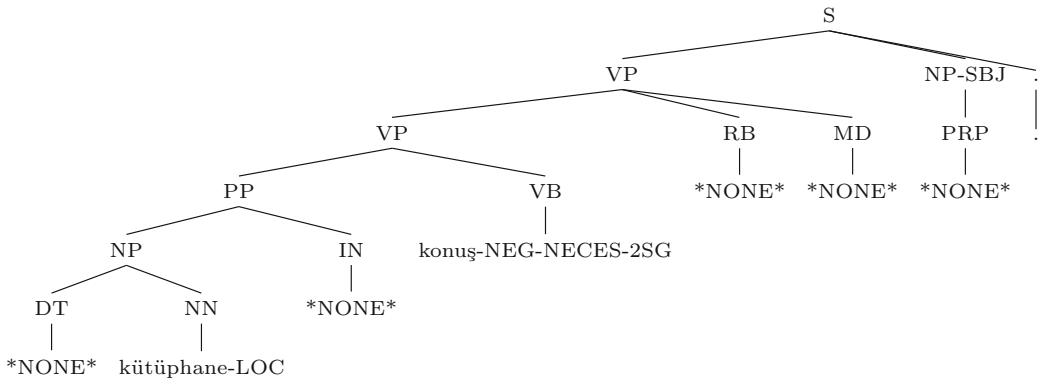
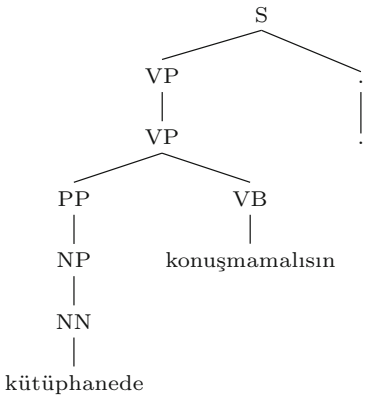


Fig. 2 Morphological analysis of the translated tree in Fig. 1

Fig. 3 After removal of *NONE* leafs



3.3.1 Removal of *NONE* Leafs

The translated trees contain *NONE* leafs vacated by English constituents embedded in the morphemes of Turkish stems. After morphological annotation, the semantic aspects of those English constituents will be represented in the morpheme leaves. Hence, we remove all *NONE* leaves and all their ancestors until we reach an ancestor that has more than one child. When we apply this rule to the translated tree in Fig. 1, we end up with the tree in Fig. 3.

3.3.2 Branching Multiword Leaves

A single English word may be translated to Turkish as a multiword expression. In such cases, we branch the multiword leaf into multiple leaves. We assign tags to the parents of the new leaf nodes according to their morphological analyses. For instance, in Fig. 4, the translation of “cancel” is a two word expression “iptal et”. Since “iptal” is a noun and “et” is a verb, their new parents are tagged as NN and VB respectively.

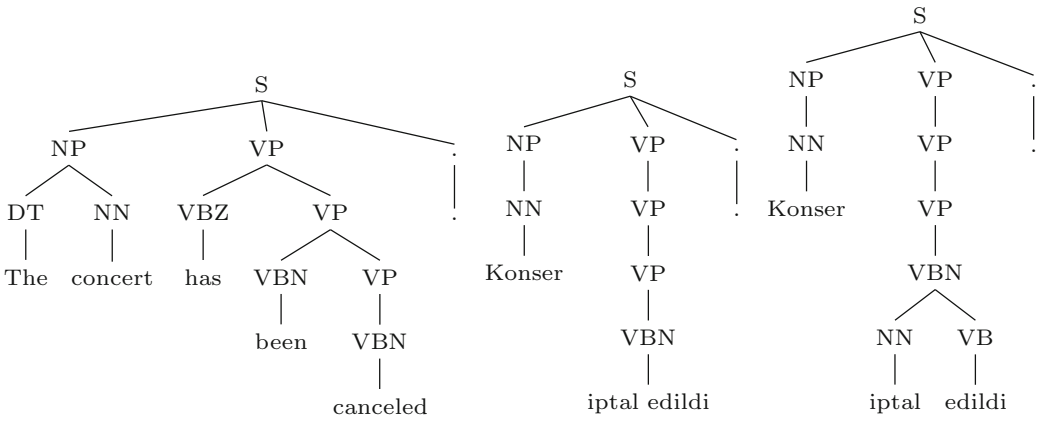


Fig. 4 Original tree, translated tree without *NONE* and tree after multiword branching konser iptal ed-il-di (concert cancel-PASS-PAST-A3SG)

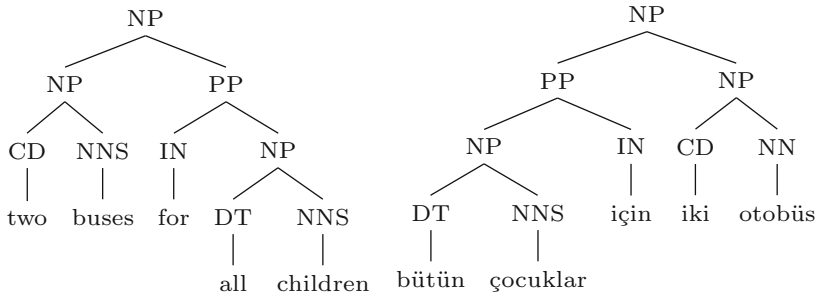


Fig. 5 Original tree and translated tree after fixing NNS tags

3.3.3 Fixing Plural Nouns

Plural nouns tagged as NNS in an English tree are sometimes translated as singular to Turkish. For example, while English nouns next to a cardinality are attached the plurality suffix, Turkish nouns are not. In such cases, we check the morphological analyses of the nouns to detect whether they have the plurality suffix "-lAr" which is equivalent to the "-s" plurality suffix in English. Since we rely on Turkish morphological analysis, irregular plural nouns of English are also tagged correctly following this rule. Figure 5 shows examples for these cases. While the translation of "children", "çocuk-lAr", contains a plurality suffix, the translation of "buses", "otobüs", does not. Therefore, we convert the NNS tag above "otobüs" to NN.

3.3.4 Removal of Unnecessary Ancestors

After removal of *NONE* leafs, we generally end up with trees that have unnecessary ancestors. For example, in Fig. 3 we have PP–NP–NN sequence and in Fig. 4 we have VP–VP–VP–VBN sequence. In the former tree NP–NN sequence and in the latter tree VP–VP–VBN sequence is unnecessary and can be removed. For each

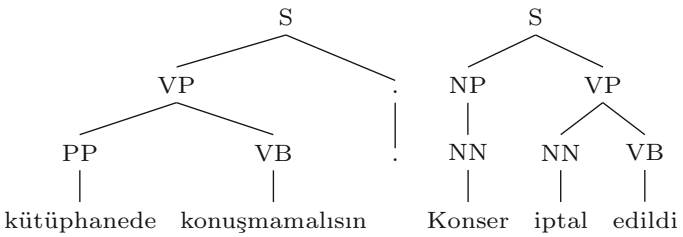


Fig. 6 Translated trees in Figs. 3 and 4 after ancestor removal

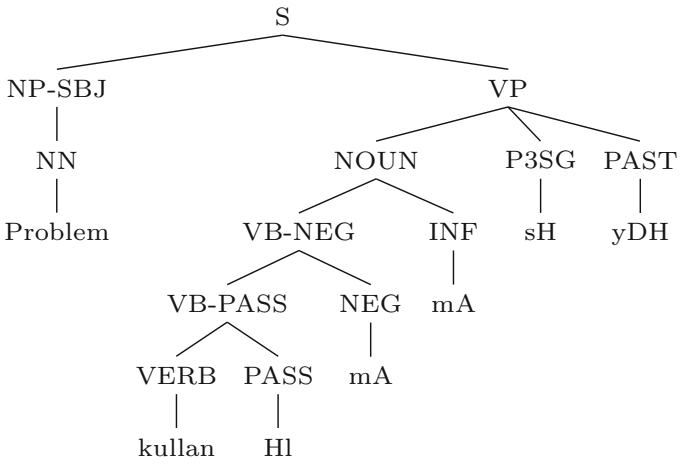


Fig. 7 Example noun suffixes problem kullan-ıl-ma-ma-sı-ydı (problem use-PASS-NEG-INF-P3SG-PAST)

node, we remove all its ancestors until we reach an ancestor that has more than one child. However, if the leaf node does not contain any suffixes, we do not remove the immediate ancestor of that node. For instance, the leaf node "Konser" in Fig. 4 does not contain any suffixes. Hence, we do not remove its parent, i.e. the NN tag, from the tree. After applying this rule, we end up with the trees in Fig. 6.

3.3.5 Branching Morphemes

The final and probably the most important operation is to branch the morphemes. We need to exploit the morphological analysis to compensate the information loss that occurred during the removal operations in the previous stages.

We treat the suffixes attached to nouns and verbs differently. While all suffixes attached to a noun will be siblings of that noun, each suffix attached to a verb will generate a new parent node and the next suffix will be a sibling of that new node. Figure 7 shows examples of both noun and verb suffixes and Fig. 8 shows examples of verb suffixes. The sentence in Fig. 7, "Problem kullanılmamasıydı" is the translation of "The problem was not to be used." As it can be seen, the suffixes attached to the nouns are siblings (i.e. P3SG and PAST) of that noun, whereas the suffixes attached to verbs create new tags (i.e. VB-PASS, VB-NEG).

In Turkish, suffixes may convert nouns to verbs or vice versa. In such cases, the corresponding node will be treated according to its final form and any additional

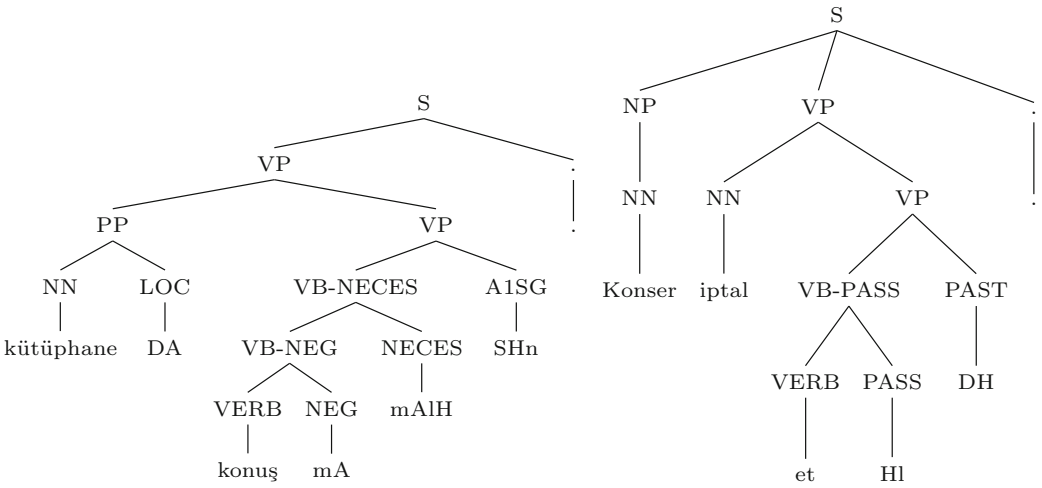


Fig. 8 Final forms of the trees in Fig. 6 after morpheme branching

suffixes will be attached following the above rule. Figure 7 shows an example of converting a verb to a noun.

In this stage, we introduced new tags for morphological annotation. Verb and noun roots are tagged as VERB and NOUN respectively. Suffixes attached to verbs determine the new tag that will be created. We simply concatenate their morphological annotation to VB-. For example, as shown in Fig. 8, VB-NECES is used when the NECES suffix is attached to a verb. Figure 8 shows final forms of the trees in Fig. 6 after morpheme branching.

4 Conclusion

We reported our preliminary efforts for building a constituency parse tree corpus for Turkish by translating and transforming a subset of the parse trees of the English Penn Treebank. We covered over 50 % of the sentences with a maximum length of 15 tokens including punctuation. We believe that this will be a useful resource for Turkish statistical NLP studies. The corpus will enable the development of statistical phrase-structure parsers for Turkish. It can also be used in English-Turkish statistical machine translation studies as a parallel treebank. As a future work, we plan to expand the dataset to include all the Penn Treebank sentences.

References

1. Abeillé, A., Clément, L., Toussenel, F.: Building a treebank for French. In: Treebanks, pp. 165–187. Springer (2003)
2. Atalay, N.B., Oflazer, K., Say, B.: The annotation process in the Turkish treebank. In: 4th International Workshop on Linguistically Interpreted Corpora (2003)

3. Brants, S., Dipper, S., Hansen, S., Lezius, W., Smith, G.: The tiger treebank. In: Proceedings of the workshop on treebanks and linguistic theories, vol. 168 (2002)
4. Cakici, R.: Automatic induction of a ccg grammar for Turkish. In: ACL Student Research Workshop (2005)
5. Cetinoglu, O., Oflazer, K.: Morphology-syntax interface for Turkish lfg. In: Computational Linguistics and Annual Meeting of the Association (2006)
6. Cetinoglu, O., Oflazer, K.: Integrating derivational morphology into syntax. In: Recent Advances in Natural Language Processing V (2009)
7. Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The szeged treebank. In: Text, Speech and Dialogue, pp. 123–131. Springer (2005)
8. Eryigit, G., Nivre, J., Oflazer, K.: Dependency parsing of Turkish. Comput. Linguist. (2008)
9. Eryigit, G., Oflazer, K.: Statistical dependency parsing for Turkish. In: 11th Conference of the European Chapter of the Association for Computational Linguistics (2006)
10. Haverinen, K., Nyblom, J., Viljanen, T., Laippala, V., Kohonen, S., Missilä, A., Ojala, S., Salakoski, T., Ginter, F.: Building the essential resources for Finnish: The Turku dependency treebank. Lang. Resour. Eval. 1–39 (2013)
11. Kornfilt, J.: Turkish. Routledge (1997)
12. Maamouri, M., Bies, A., Buckwalter, T., Mekki, W.: The penn Arabic treebank: Building a large-scale annotated Arabic corpus. In: NEMLAR Conference on Arabic Language Resources and Tools, pp. 102–109 (2004)
13. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of english: The penn treebank. Comput. Linguist. **19**(2), 313–330 (1993)
14. Oflazer, K.: Two-level description of Turkish morphology. Literary Linguist. Comput. **9**(2), 137–148 (1994)
15. Riedel, S., Cakici, R., Meza-Ruiz, I.: Multi-lingual dependency parsing with incremental integer linear programming (2006)
16. Ruket, C., Baldridge, J.: Projective and non-projective Turkish parsing. In: Fifth International Workshop on Treebanks and Linguistic Theories (2006)
17. Xue, N., Xia, F., Chiou, F.D., Palmer, M.: The penn Chinese treebank: Phrase structure annotation of a large corpus. Nat. Lang. Eng. **11**(2), 207–238 (2005)
18. Yıldız, O.T., Solak, E., Görgün, O., Ehsani, R.: Constructing a Turkish-English parallel treebank. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 112–117. Association for Computational Linguistics, Baltimore, Maryland (2014)
19. Yuret, D.: Dependency parsing as a classification problem. In: Proceedings of the Tenth Conference on Computational Natural Language Learning (2006)