Creating A Syntactically Felicitous Constituency Treebank For Turkish

Neslihan Kara
Starlang Yazılım Danışmanlık
İstanbul, Turkey
neslihan@starlangyazilim.com

Büşra Marşan
Starlang Yazılım Danışmanlık
İstanbul, Turkey
busra@starlangyazilim.com

Merve Özçelik
Starlang Yazılım Danışmanlık
İstanbul, Turkey
merve@starlangyazilim.com

Bilge Nas Arıcan
Starlang Yazılım Danışmanlık
İstanbul, Turkey
bilge@starlangyazilim.com

Aslı Kuzgun
Starlang Yazılım Danışmanlık
İstanbul, Turkey
asli@starlangyazilim.com

Neslihan Cesur
Starlang Yazılım Danışmanlık
İstanbul, Turkey
nesli@starlangyazilim.com

Deniz Baran Aslan
Starlang Yazılım Danışmanlık
İstanbul, Turkey
deniz@starlangyazilim.com

Olcay Taner Yıldız
Işık University
İstanbul, Turkey
olcaytaner@isikun.edu.tr

***Abstract*—In this study, Bakay et. al [1] and Yıldız et. al.'s [2] work on Turkish constituency treebanks were developed further. Compared to the previous work, the most prominent feature of this study is the fact that every annotation and refinement process is held manually. In addition, constituency treebank created as a result of this study abides by the syntactic rules and typologic features of Turkish while the trees created by previous studies convey only the translated and simply inverted trees that completely ignore the syntactic properties of Turkish. The methodology followed in this study resulted in a significantly more accurate representation of Turkish language and simpler, relatively flatter trees. The straightforward style of trees in this study reduces the complexity and offers a better training dataset for learning algorithms.**

***Index Terms*—Constituency, TreeBank, Constituency Parsing**

I. Introduction

When one talks about the "success" of a Natural Language Processing solution, they often refer to its ability to analyse the semantic and syntactic structure of a given sentence. Such a solution is expected to be able to understand both the linear and hierarchical order of the words in a sentence, unveil embedded structures, illustrate syntactical relationships and have a firm grasp of the argument structure. In order to meet the expectations, cutting edge Natural Language Processing systems like parsers, POS taggers or machine translation systems make use of syntactically or semantically annotated treebanks. Such treebanks offer a deep look through the surface and into the logical form of sentences.

Annotated treebanks can be categorised as constituency treebanks and dependency treebanks. The former offers clarity through resolving structural ambiguities, and successfully illustrates the syntagmatic relations like adjunct, complement, predicate, internal argument, external argument and such. While the latter offers a fast and flexible semantic analysis.

The very first comprehensive annotated treebank, the Penn Treebank, was created for the English language and offers 40,000 annotated sentences [3]. Following the Penn Treebank, numerous treebanks annotated for constituency structures were developed in different languages including French [4], German [5], Finnish [6], Hungarian [7], Chinese [8] and Arabic [9].

978-1-7281-9136-2/20/$31.00 © 2020 IEEE

In this study, the aim is to create a unique and coherent constituency treebank for Turkish language by following the footsteps of the Penn Treebank, and elaborating on the previous work by Bakay et al. [1] and Yıldız et al. [2].

The main goal of this study was to come up with a coherent treebank that is able to express typological particularities of Turkish, such as its highly agglutinative and rich morphology and flexible word order. Different from most Indo-Eurpoean languages like English, Turkish employs suffixes as appositions, marks most internal arguments and adjuncts with case (accusative, dative, ablative and such) [10]. Moreover, Turkish also diverges from its standard Subject-Object-Verb word order to stress and/or topicalize certain words or phrases [11]. As a result, directly "translating" an English treebank fails to capture the characteristics of Turkish language and produces many empty projections in the target structure. That is why utmost attention was paid to create corresponding structures in Turkish instead of merely inverting the Penn Treebank trees and changing the leaf nodes.

A significant portion of this study was built on Bakay et al. and Yıldız et al.'s previous work on constituency parsing. Manually translated parse trees were used, automatic morphological annotation process was carried using a morphological analyser [12], empty projections and unary branches were deleted, tags were refined by introducing new ones and removing the irrelevant ones for Turkish, and finally, a team of annotators polished the trees by manually going through every single one for coherence and consistency. This fastidious approach for creating the dataset has resulted with more accurate representations in terms of capturing the richness of Turkish, a rather a complex language to parse.

This paper is organised as follows: Previous Work on Constituency Parsing where the current literature is discussed, The Journey from English Penn Treebank to Turkish Constituency-Parsed Treebank where the steps of creating a new constituency treebank is explained, Results and Conclusion.

II. Previous Work on Constituency Parsing

After the Penn Treebank has set the international gold standard for annotated treebanks, similar works have been conducted to create annotated treebank corpora in different languages including German [5], French [4], Chinese [8], Finnish [6], Hungarian [7], Arabic [9] and Turkish.

The very first Turkish treebank was created in 2003 by scholars from METU and Sabancı. Titled METU-Sabancı Dependency Treebank, this work covers manually annotated 7,262 sentences. The annotation process of this treebank addressed head-dependent relations, morphological information and syntactic categories. Due to being the first of its kind, METU-Sabancı Treebank has gained a significant recognition and been referred to in numerous studies on dependency parsing [13] [14] [15] and various other Natural Language applications [16] [17] [18], [19]. Yet it was only in 2015 that a comprehensive Turkish treebank corpus was created through phrase-structure annotation [2]. In this study, 5,143 trees from Penn Treebank II were manually translated, then the morphologic annotation was done automatically. Finally, parse trees were refined automatically through the implementation of the manually designed rules.

Following Yıldız et al.'s study [2], another attempt at creating Turkish trees from Penn Treebank II data was done by Görgün et al., [20] and a translation-based approach was taken by Bakay et al [1]. Yıldız et al.'s and Bakay et al.'s studies are discussed and how this study diverges from their methodology is explained in the following sections.

A. A Translation-Based Approach

In their work, Bakay et al. [1] derive their data from Penn TreeBank. For the study, 10.000 treebank sentences were translated into Turkish and went through morphological disambiguation and refinement process. Afterwards, they were subjected to shallow parsing and edited into Turkish constituency parse trees to conduct a tree based translation process [1]. Since Turkish is a free word order language, the distance between syntactically dependent parts can be longer than that in fixed word order languages. An example of this phenomenon is illustrated in Figure 1.

Bayan Marşan'ın üzerinde yıldızlar olan kırmızı kapaklı defterini buldum.
I found Ms. Marşan's notebook that has a red cover with stars on it.

Fig. 1. Long distance-relationship in Turkish vs English

Corresponding parts in this sentence are colour-coded. It can be noticed easily that the words composing the phrase "Ms. Marşan's notebook" are adjacent in English whereas in Turkish, the phrase is separated into two parts due to the intervening adjectives which define the word "notebook". This type of long distance relationships are frequently observed in Turkish, a textbook example of agglutinative languages.

To deal with such long-distance dependency relationships in machine translation from English into Turkish, Bakay et al. used a tree-based approach and successfully achieved higher BLEU scores than earlier works. Yet their method of creating trees is far from complying with the syntactic rules and limitations of Turkish. A prime example of how their preferred method errs can be seen in Figure 3. In this tree, the predicate "garantili" is tagged as VBN (verb, past participle) yet it clearly is a nominal.

B. An Automatic Translation Process

Yıldız et. al [21] proposes an automatic translation process for the Penn Treebank data. In their study, the process started with the manual translation of the trees which is enhanced by using a tool that offers the statistically most likely translation within the given context. During the translation process, some of the English constituents (represented as leaves in the tree) were embedded into Turkish morphemes (represented as morphemes appended to a single constituent in a leaf) due to the morphological differences between Turkish and English.

Some function words in English correspond to (often bound) morphemes in Turkish. Person and number agreement on the predicate is a fine example of this phenomenon and can be seen in Figure 1. In this sentence, the predicate "buldum" has the first person singular marker, "-m" on it. Since Turkish is a pro-drop language, the subject is not overt in the sentence; hence "I found" corresponds to "bul+du+m." Therefore, it is inevitable that the leaf for "I" in English will be empty when the tree is merely translated into Turkish. For empty leaves, the expression *NONE* is used in the trees of Yıldız et al.'s 2014 study [21].

In their 2015 study [2], Yıldız et al. have gotten rid of *NONE*s. They created a constituency parse treebank, annotated it automatically and refined it manually. For the corpus, only the sentences with 15 or less tokens were selected and 9560 trees were created. Before shallow-parsing and POS tagging, trees were morphologically annotated using an automatic analyser based on 1994 work of Oflazer [22]. Then annotators manually chose the correct meaning from a set of meanings generated automatically for each morpheme in each word. The reason for implementing the manual morphological annotation process was due to the complex morphology of Turkish where person, number, time, aspect, modality, plurality or any other information is conveyed through affixes. Moreover, some derivational affixes have the same surface representations with the aforementioned affixes. For instance, in "bil-in-dik" the suffix -dik carries the meaning from know to well-known whereas in git-tik-i the suffix -dik carries the meaning of the conjunction that (the place that he/she went). Therefore, fully automatic processes for morphological disambiguation in Turkish can be more challenging than it seems.

To refine the trees, leaves with *NONE* expression were deleted. Removing a *NONE* leaf extended to the next ancestor that had more than one child. Some single word expressions in English may correspond to multiword expressions in Turkish. In this case, two words would be represented under one node. To prevent this, multiword expressions were branched once more so that every word was represented as an individual leaf. Then each leaf got its own tag to retrieve

the morphological information. Since morphemes can change a word's syntactic class (from noun to verb, from noun to adjective etc.) the node names for the leaves were given accordingly.

For the POS tag annotation of Bakay's work, Topsakal et al.' work [23] was taken as the baseline. Topsakal et al. created a parser that gives sufficient information about the syntactic segments of a sentence, which is a rare study topic for Turkish because of its agglutinative nature. In Turkish shallow parser, there are 5 main parts of a sentence: Özne (Subject), Yüklem (Predicate), Nesne (Direct Object), Zarf Tümleci (Adverbial Clause) and Dolaylı Tümleç (Oblique Object). There are no hierarchical structures in parsed outcome, so a bracketed system is enough for a shallow parse. Following the steps above, Bakay and her team used the morphologically analysed trees to get a higher accuracy in their tree-based translation approach.

The processes in previous studies were mostly carried out automatically. On the other hand, the annotation, refinement and control processes to establish inter-annotator agreement in this study were carried out fully manually. Moreover, a precise morphological analyser is used [12]. As a result, a treebank that captures the particularities of the Turkish language is created. Due to its accuracy and consistency, this treebank can also be used in other NLP tools, studies and processes.

III. The Journey from English Penn Treebank to Turkish Constituency-Parsed Treebank

A. Pre-Annotation Process

After the manual translation process; morphological analysis was carried out and empty projections were deleted. Next, a team of annotators who are also the authors of this article went through the trees for the sake of refinement, inter-annotator agreement and coherence.

First, the team of annotators assessed the quality of the translation since some of the expressions were translated inaccurately even though the translation was done by professional translators. After proofreading and editing the Turkish translations, the team discussed and decided on employing a non-binary branching style. Non-binary branching was used to create a levelled representation that can also go one level deeper to illustrate a particular syntactic relationship. Also this representation method allowed gathering all specifiers in one level.

For instance in a phrase like "Guaranteed by Dai-Ichi Kangyo Bank Ltd.", non-binary branching allows gathering all nouns of the phrase "Da-Ichi Kangyo Bank Ltd" under the same NP (noun phrase). This methodology precludes creating a spec-NP level for each of the 4 words in the phrase. To express transitivity of a verb which requires bar level representations for the objects, binary branching was employed. Adverbial phrases, post positions and similar structures were kept at the same level. The verb head and the object (internal argument) were kept adjacent in one level below.

As illustrated in Figure 4, English verb "to guarantee" is translated into multi word expression "garanti vermek" in Turkish. The same logic as transitivity in multiword expressions was applied since they represent a single phrase.

By-phrase in English is represented as a postpositional phrase with the head "tarafından" in Turkish since it dominates the preceding NP, the company name is given under another separate NP.

In the trees created for this study, the head of the whole sentence is S (for "sentence") and it projects an NP for subject and a VP (verb phrase) for the predicate by default, although not all the predicates are verbal and not all the subjects are nominal roots per se. These issues will be discussed in the following sections. In Figure 4, there is not a noun phrase in the subject position since the sentence is in passive voice. So this sentence only has a VP modified by a PP and a punctuation which has its own branch. The punctuation marks are kept where they were in Penn Treebank trees.

B. The Annotation Process

After reviewing the sentences, the team of annotators decided on changing, adding or removing some tags to represent the syntactic properties of Turkish more accurately.

Predicates were revised first. If the morphological root of a predicate given in the shallow parse was not a verb then the predicate was tagged as NOMP (nominal phrase), anything else was left as VP. As for the objects of the verb, if the morphological root of an object or a subject was not defined as a noun by the shallow parser, the mother node was kept as an NP but the verbal structure was represented under the mother node tagged NP. For instance, the sentence "Sentaksı sevmek kaçınılmazdır." *("Loving syntax is inevitable.")* is parsed and tagged as follows:

S -> NP[Sentaksı sevmek] NOMP[kaçınılmazdır] PUNCT[.]
NP -> NP[Sentaksı] VP[sevmek]
VP -> NOMP[kaçınılmazdır]
PUNCT -> .[.]

Thus, the internal structure of a phrase is preserved regardless of the position of the phrase (subject, predicate, object etc.). If the predicate of the clause had verbal agreement and was finite, the maximal projection of such phrases were designated as another S. For example, the subordinate clauses formed using "that" are represented as another S in Turkish trees since they have a [S -> NP VP] structure.

In order to keep their internal structure intact, the multiword expressions like "garanti vermek" *("to guarantee")* were branched in two under a maximal VP projection:

VP -> NP[garanti] VP[vermek]

In a sentence like "I am not a doctor." the predicate is [am not a doctor]. Since the predicate is non-verbal, such sentences are represented using the NOMP tag. The negation morpheme is tagged NEG. "Not" is translated into Turkish as "değil," so all the "değil" expressions are tagged NEG under NOMP. In such cases, NEG is a sister to the predicate and they have a C-command relationship. Only the non-verbal predicates take "değil" negation whereas negated verbal predicates take

the suffix -ma, thus NEG leaf is only present in NOMP projections.

Verbs can also take question particles. In Turkish, yes no questions are formed by "mI" clitic. Since this clitic is an orthographic word governed by the VP (predicate), all instances of it were tagged as QP (question particle) since it carries some information like person and time. For wh-questions, WP tag was used. When the same expression indicated a relative clause such as "I don't know where he went." *("Nereye gittiğini bilmiyorum"),* the appropriate tag was used.

For interjections, INTJ tag was used.

For post positions, PP tag was used and represented as head of the phrase they are in.

For conjunctions including the morpheme "de" in Turkish, CONJP tag was used. They were right under the head S if they connected two sentences. Otherwise, conjunctions were placed in their subtrees.

For numerical expressions, NUM (number) was used. When such expressions were marked with case or represented a ratio like the word "percent," NP tag was used.

The determiners were marked as DT. The most common usage was "a/an" *("bir").* In Turkish, determiner "bir" can be attached to adjectival phrases like "herhangi" *("any")* "bir şey" *("something")* or "birçok" *("most").* It was taken as DT in all cases. The word "bir" itself can represent the number "one", as well. In that case, it was tagged as NUM. Another common determiner is bu/şu/o" *("this/that")* which can be considered as an NP unless it precedes a noun.

After deciding on all of the rules and tags, all sentences were edited accordingly. A graphic user interface was used. This interface also showed the original English sentence, its Turkish translation and the tree representation of the Turkish translation. It allowed going back and forth between trees, adding a branch to the tree or a subtree, removing a branch or subtree from the tree or a subtree, renaming the tag or switching positions of two nodes. Since the interface allowed viewing any tree in a given folder, the annotators were able to could see each others' annotations and discuss when needed.

1) Inter-Annotator Agreement: After the annotation process was done, a member of the annotator team took the role of the controller. She went through all the annotations and reported the issues she identified. According to her report, inter-annotator agreement was mostly very high but there were some cases where the annotators had different takes on some tags. For instance, the adjective was sometimes tagged as NP in some adjectival phrases but it must be tagged as ADJP (adjectival phrase). The controller converted expressions like "%10" from [NUM NUM] to [NP NUM] since the percent sign has an NP tag. She also noted that some instances of negation "değil" were incorrectly tagged as NP and she changed such instances into NEG. Another issue was about the determiners "bir" and "bu". Some determiner "bir"s and "bu"s were tagged as ADJP or NP which were later changed to DP by our controller. She noted that non-verbal predicates were often tagged with VP which is incorrect. She corrected such occurrences by tagging them as NOMPs. It can be inferred from her report that even when the annotator team had conflicting opinions, they used relatively similar tags. So the incorrect annotations weren't completely wrong per se but they just didn't fit framework that was created during the pre-annotation process. To ensure the inter-annotator consistency, the controller fixed all the inconsistent cases.

C. Challenges Encountered During the Annotation Process

Mostly because of the morphological structure of Turkish, the annotator team had some conflicting opinions during the annotation process. For instance, some structures have more than one function and each function may require its own tag. The morpheme "dA" is a good example of this phenomenon: It can be used as the locative case, as a conjunction and as a morpheme indicating "inclusiveness". In the first case, it was not tagged because it is already attached to a noun in its locative form. As for the second and third cases, it was hard to decide which tag to use for the appropriate representation of "dA". After heated discussions, it was decided that both cases should be tagged as CONJP.

Another challenge was the question particles in Turkish. Since English has no overt question markers or particles, it was necessary to come up with an appropriate and unique tag for these instances in Turkish, thus the annotator team decided on creating QP (Question Particle) tag. It must be noted that in Penn Treebank, QP tag exists for quantifier phrases that define complex amounts and/or measures. The NUM tag was used for this purpose and the QP tag was employed for question particles in the treebank.

IV. Results

In this study, annotated trees follow a minimalistic approach in a sense that the number of branches are minimized as much as possible since an additional level of projection add an unnecessary layer of complexity to the trees. As a result, the number of daughter branches are significantly lower in the Turkish trees than in their English counterparts (see Figure I). The number of a maximal projection XP, such as NP, VP, ADVP, ADJP, NOMP etc., that has only one daughter is 117,568 in English trees while it is as low as 81,613 in the Turkish trees. The main reason for this contrast is the fact that unary branches (or bar levels, as referred to in X-bar theory) that project NONE, VBN, NNP, IN, NSBJ-1 and such tags are eliminated in the annotation process of this study. Although Turkish diverges from English in having a significant amount of multi word expressions, this decision led to a lower average number of branches per sentence in Turkish: On average, Turkish trees have 13.62 nodes while English trees have 18.56 nodes.

When the sentences were first translated from English Penn TreeBank into Turkish, there were empty projections whose leaves were replaced with *NONE*. The reason behind this phenomenon was the fact that Turkish employs suffixes and case markers where English uses appositions and individual words. Previous studies on Turkish constituency trees [2] [1]

TABLE I
THE NUMBER OF DAUGHTER BRANCHES AND THEIR FREQUENCIES

Daughter Branch	English	Turkish
1	117568	81613
2	38068	31869
3	15242	12316
4	3954	2976
5	1787	1067
6	465	288
7	269	102
8	21	7
9	8	5
10+	4	2
Average # of Branches per Sentence	18.56	13.62

kept these nodes (see Figure 2 and Figure 3). In this study, all instances of *NONE* are deleted.

First attempts to create semi or fully automatic constituency treebanks from Penn TreeBank sentences had resulted with trees that have many unnecessary, almost obsolete branches and nodes due to keeping additional tags like NNP, VBN, NP-LGS and IN (Figure 2 and 3). Due to the methodology, manual translation and annotation process followed in this study, the number of nodes have dropped significantly (Table I), thus the treebank offers a clear, minimalistic view and representation that still can illustrate the complexity of the sentences while missing out none of the critical information (Figure 4).

On average, this treebank draws 5 less nodes for each sentence[1]. Since the binary branching rule was not adopted, trees and subtrees have all the adjuncts at the same level and if there is a complement, the complement itself along with the constituent it depends on goes one level deeper in the tree. Thus a flatter but equally, if not more, efficient and coherent representation is created.

Contrary to the previous studies in Turkish, this study adopts manual annotation processes. Thus the procedure results with trees adapted for Turkish. The approach in this study has intrinsic value for capturing the particular linguistic features of Turkish such as its headedness, rich morphology, flexible word order, and even its syntactically ambiguous structures. This is an important advantage since the treebank created by our processes can be used to train more advanced NLP algorithms and create more precise syntactic parsers.

In the Table II, the difference in tag frequency between the Turkish trees and their source English trees can be seen clearly. Aside from the introduction of new tags like NOMP, the difference between NP, PP, ADJP, DT and CONJP counts are striking. This explicit contrast in numbers clearly illustrates the typological differences between the two languages. Moreover, the introduction of new tags like NEG, NOMP or QP for question particle makes this treebank somewhat different from the Penn Treebank while maintaining a certain level of correspondence.

A chronologically aligned comparison of the previous Turkish constituency trees and the ones created in this study can be seen in Figure 2, Figure 3 and Figure 4.

[1]https://github.com/olcaytaner/TurkishAnnotatedTreeBank-15

TABLE II
PHRASE TAGS AND THEIR FREQUENCIES

Tag Name	English	Turkish
S	11583	10367
NP	32457	56854
VP	15863	15896
NOMP	0	4743
PP	7291	4077
ADJP	1876	9795
ADVP	2957	5780
DT	7876	3191
NUM	0	4310
QP (Quantifier Phrase)	1005	0
QP (Question Particle)	0	73
CC	1846	0
CONJP	7	2441
NEG	0	240

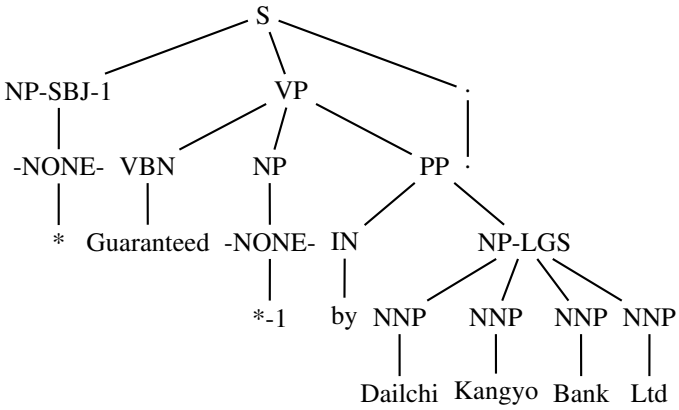

Fig. 2. A sample constituency tree generated by Prof. Yıldız's old methodology

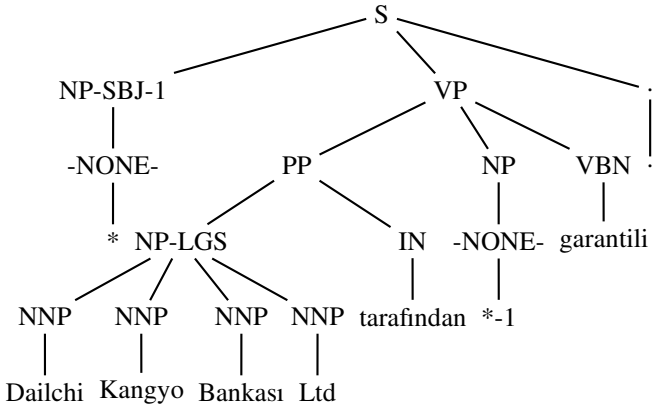

Fig. 3. A sample constituency tree generated by Bakay's methodology

V. CONCLUSION

A. The Significance of This Study

Previous work on Turkish constituency treebanks offer merely translated sentences with empty projections. On the other hand, this study presents a constituency treebank that is adapted to correspond the typological features of Turkish. Parsing and annotation rules were created to capture the complexity and rich morphology of Turkish. As a result, this study is able to offer a coherent constituency treebank that directly corresponds to Penn Treebank but manages to illustrate particularities of Turkish and consequently, be useful

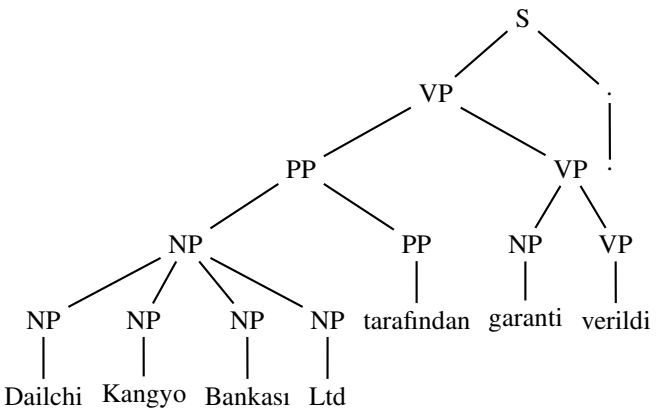

Fig. 4. A sample constituency tree generated the methodology followed in this study

as a standalone constituency treebank as well.

Moreover, the constituency in this study don't have empty projections, are relieved of redundant unary branches, and can distinguish syntactic roles of adjuncts and arguments from one another. As a result, TRopBank [24] can also be integrated in order to enhance the constituency treebank with a layer of semantic information. A similar work has been conducted by Kingsbury, Palmer and Marcus [25] for English.

Finally, this treebank offers a training dataset that is significantly less noisy for we have manually annotated each tree and omitted unary branches along with empty projections. As a result, the treebank can be used in context-free grammar applications, training syntactic parsers or dependency-constituency conversion algorithms and creating machine translation systems.

B. Suggestions for Further Study

In our work, we opted for distinguishing nominal predicates and verbal predicates since Turkish licenses the use of a nominal predicate without an auxiliary verb, copula or a light verb. Whether making the distinction between VP and NOMP is useful in training parsing algorithms is not certain for now. Further study must be conducted regarding the benefits of such a distinction.

As mentioned above, Turkish licenses topicalization for stressing certain phrases and/or words in accordance with the context. In theoretical linguistics, derivational theories of syntax refer to syntactic movement [26] in order to illustrate non-canonical word order (any order that deviates from Subject-Object-Verb) in Turkish [27]. In our constituency trees, we disregarded movement and consequently, co-referencing (See [28]). As a result, some of the sentences with non-canonical word order fail to express binding principles and traditional hierarchical order of internal and external arguments. A way to illustrate non-canonical word order while preserving the hierarchy is a challenge that remains unconquered. Further study might discover a new way to illustrate movement or non-canonical word order.

In addition, the Penn Treebank sentences have a strictly formal style and are excerpts from financial articles. Introducing additional data from various sources like daily speech or literary texts can be very beneficial for providing challenging test data and/or training sets for machine translation systems, syntactic parsers and similar NLP applications.

REFERENCES

[1] O. Bakay, B. Avar, and O. T. Yildiz, "A tree-based approach for English-to-Turkish translation," *Turkish Journal Of Electrical Engineering And Computer Sciences*, vol. 27, pp. 437–452, 01 2019.

[2] O. T. Yildiz, E. Solak, S. Candır, E. Ehsani, and O. Gorgun, "Constructing a Turkish constituency parse treebank," vol. 363, 01 2015, pp. 339–347.

[3] M. Marcus, M. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of English: The penn treebank," *Computational Linguistics*, vol. 19, pp. 313–330, 07 2002.

[4] A. Abeill, L. Clment, and A. Kinyon, "Building a treebank for french," 10 2000.

[5] S. Brants, S. Hansen, and W. Lezius, "The tiger treebank," 11 2002.

[6] K. Haverinen, J. Nyblom, T. Viljanen, V. Laippala, S. Kohonen, A. Missilä, S. Ojala, T. Salakoski, and F. Ginter, "Building the essential resources for Finnish: the Turku dependency treebank," *Language Resources and Evaluation*, vol. 48, 09 2013.

[7] D. Csendes, J. Csirik, T. Gyimóthy, and A. Kocsor, "The szeged treebank," in *Proceedings of the 8th International Conference on Text, Speech and Dialogue*, ser. TSD'05. Berlin, Heidelberg: Springer-Verlag, 2005, p. 123–131.

[8] N. Xue, F. Xia, F.-d. Chiou, and M. Palmer, "The penn Chinese treebank: Phrase structure annotation of a large corpus," *Nat. Lang. Eng.*, vol. 11, no. 2, p. 207–238, Jun. 2005. [Online]. Available: https://doi.org/10.1017/S135132490400364X

[9] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki, "The penn Arabic treebank: Building a large-scale annotated Arabic corpus," *NEMLAR Conference on Arabic Language Resources and Tools*, 01 2004.

[10] J. Kornfilt, *Turkish*. Routledge, 1997.

[11] A. Goksel and O. A. Sumru, "Is there a focus position in Turkish," *Studies on Turkish and Turkic languages*, p. 107, 2000.

[12] O. T. Yildiz, B. Avar, and G. Ercan, "An open, extendible, and fast Turkish morphological analyzer," 09 2019.

[13] G. Eryigit, J. Nivre, and K. Oflazer, "Dependency parsing of Turkish," *Computational Linguistics*, vol. 34, p. 627, 12 2008.

[14] G. Eryigit and K. Oflazer, "Statistical dependency parsing for Turkish," 01 2006.

[15] S. Riedel, R. Çakıcı, and I. Meza-Ruiz, "Multi-lingual dependency parsing with incremental integer linear programming," in *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*. New York City: Association for Computational Linguistics, Jun. 2006, pp. 226–230. [Online]. Available: https://www.aclweb.org/anthology/W06-2934

[16] R. Cakici, "Automatic induction of a ccg grammar for Turkish," 01 2005.

[17] D. Yuret, "Dependency parsing as a classification problem," pp. 246–250, 07 2006.

[18] O. Cetinoglu and K. Oflazer, "Morphology-syntax interface for Turkish lfg," 07 2006.

[19] ——, *Integrating Derivational Morphology into Syntax*, 01 2009.

[20] O. Gorgun, O. T. Yildiz, E. Solak, and E. Ehsani, "English-Turkish parallel treebank with morphological annotations and its use in tree-based smt," 01 2016, pp. 510–516.

[21] O. T. Yildiz, E. Solak, O. Gorgun, and E. Ehsani, "Constructing a Turkish-English parallel treebank," 06 2014.

[22] K. Oflazer, "Two-level description of Turkish morphology," vol. 9, 01 1994.

[23] O. Topsakal, O. Acikgoz, A. Gürkan, A. B. Kanburoglu, B. Ertopcu, B. Ozenç, I. Cam, B. Avar, G. Ercan, and O. T. Yildiz, "Shallow parsing in Turkish," 10 2017, pp. 480–485.

[24] N. Kara, B. Marsan, D. Aslan, O. Bakay, K. Ak, and O. T. Yildiz, "Tropbank: Turkish propbank v2.0," 05 2020.

[25] P. Kingsbury, M. Palmer, and M. Marcus, "Adding semantic annotation to the penn treebank," *Proceedings of HLT*, 04 2003.

[26] G. Graffi, *200 Years of Syntax: A critical survey*. John Benjamins, 2001.

[27] L. Haegeman and J. Gueron, *English Grammar: A Generative Perspective*. Wiley, 1998.

[28] C. Pollard and I. Sag, "Anaphors in english and the scope of binding theory," *Linguistic Inquiry*, vol. 23, 01 1992.