SYNTACTIC PARSING APPROACHES FOR ENGLISH, TURKISH AND UZBEK

Rakhmonova Mohira Abdusattor qizi
State University of Uzbek Language and Literature, Tashkent, Uzbekiston
mokhirata@gmail.com

Abstract. This article discusses about syntactic parsing methods and related works on dependency Parsing. There are transition-based and graph-based approaches applied to dependency parsing problem in the literature. As well as the constituency parse tree is based on the formalism of context-free grammars. In this type of tree, the sentence is divided into constituents, that is, sub-phrases that belong to a specific category in the grammar.

Keywords: *parsing, dependency parsing, constituency parsing, transition-based approaches, graph-based approaches, parse tree, context-free grammars.*

Syntactic parsing is the automatic analysis of syntactic structure of natural language and the task of assigning a syntactic structure to a sentence, especially syntactic relations (in dependency grammar) and labelling spans of constituents (in constituency grammar).[1] It is motivated by the problem of structural ambiguity in natural language: a sentence can be assigned multiple grammatical parses, so some kind of knowledge beyond computational grammar rules are need to tell which parse is intended. Syntactic parsing is one of the important tasks in computational linguistics and natural language processing, and has been a subject of research since the mid-20th century with the advent of computers.[18]

There are two types of parsing methods:

1. Dependency Parsing
2. Phrase Structure Parsing (Constituency Parsing) [3]

Dependency Parsing is the task of finding the grammatical structure of a sentence by identifying the syntactic and semantic relationships between words. Dependency parsing has been utilized in many other NLP tasks such as machine translation [4, 5], relation extraction [6, 7], named entity recognition [8, 9], information extraction [10, 11], all of which involve natural language understanding to an extent. Each dependency relation is identified between a head word and a dependent word that modifies the head word in a sentence. Although such relations are considered as syntactic, they are naturally built upon semantic relationships between words. For example, each dependent has a role of modifying its head word, which is a result of a completely semantic influence. Dependency structures are represented either by hierarchical structures which are called dependency trees, or represented in the form of directed graphs, which are called dependency graphs. An

example dependency graph for the sentence Thank you, Mr. Pottering. is given below:

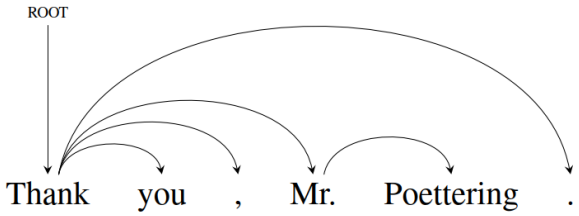

Figure 1. An example of dependency parsing for the sentence “Thank you, Mr. Poettering.” [12]

Syntactic relations between words are generally figured out with an arrow in a dependency tree, which connects each head word to a dependent. In other words, in a relation such as I → am; “I” becomes the head and “am” becomes the dependent and the arrow between them states a dependency between the two words. The ROOT token represents the root of the dependency tree (i.e. the starting point of dependency parsing or the head of the complete sentence). Even if the rules of dependency parsing will be discussed later, it is good to state here that every sentence must contain a ROOT token in its dependency tree. Dependency parsing is a task that finds the lexical dependencies between words in a sentence, and thereby extracts the grammatical structure of a sentence. Dependency is a head-dependent relation between the words. The head is the one that affects the dependents, and in other words each dependent is affected by its head. The arrows in dependency trees are always from the head to the dependents. There are two main approaches applied to dependency parsing problem in the literature: transition-based and graph-based.[12]

Transition-based approaches are generally based on transition commands and a two-stack structure that contains a dependency stack and a word buffer. Word buffer contains the words in a sentence. Words are drawn from the word buffer and pushed into the dependency stack. If there is a transition between the top two words of the dependency stack, then a dependency is created between them and this operation continues until there are no words in the dependency stack. The last word in the dependency stack would be the ROOT, which is the root of the dependency tree; starting point of the whole dependency parsing process.[12]

Graph-based approaches are generally based on performing the entire parsing process as graph operations where the nodes in the graph represent the words in a sentence. For the sentence, “John saw Mary”, imagine a weighted graph G with four vertices where each of them refers to a word including the ROOT. Edges store the dependency scores between the words.

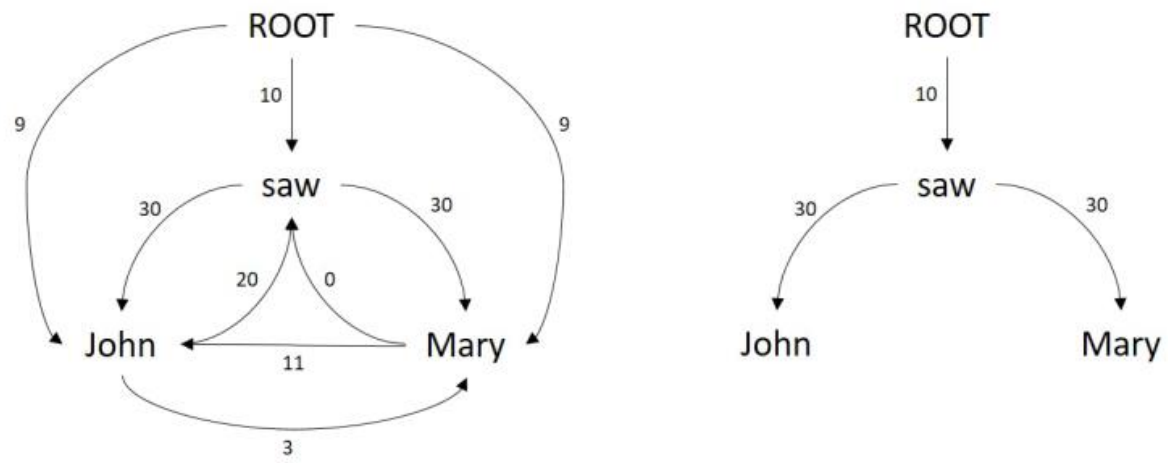

Figure 2. Graph-Based Dependency Parsing [12]

Related Work on Dependency Parsing

Eryigit and Oflazer (2006) [14] come with the idea of using inflectional groups (IGs) for dependency parsing. In their study, the authors use a statistical parser that firstly computes unit-unit relations where the units are words or IGs and then finds the maximum spanning tree from these computed relations. They have three baseline models: Word-based, IG-Based, and IG-Based with word-final IG contexts which is an IG-Based model with strict outputs. As expected, IG-Based models give the best results.

Eryigit, Oflazer and Nivre [13] show that the morphological structure plays a crucial role in Turkish dependency parsing. The authors show that parsing a sentence considering the IGs, which are sublexical units of a word, outperforms dependency parsing based on word tokens of sentences.

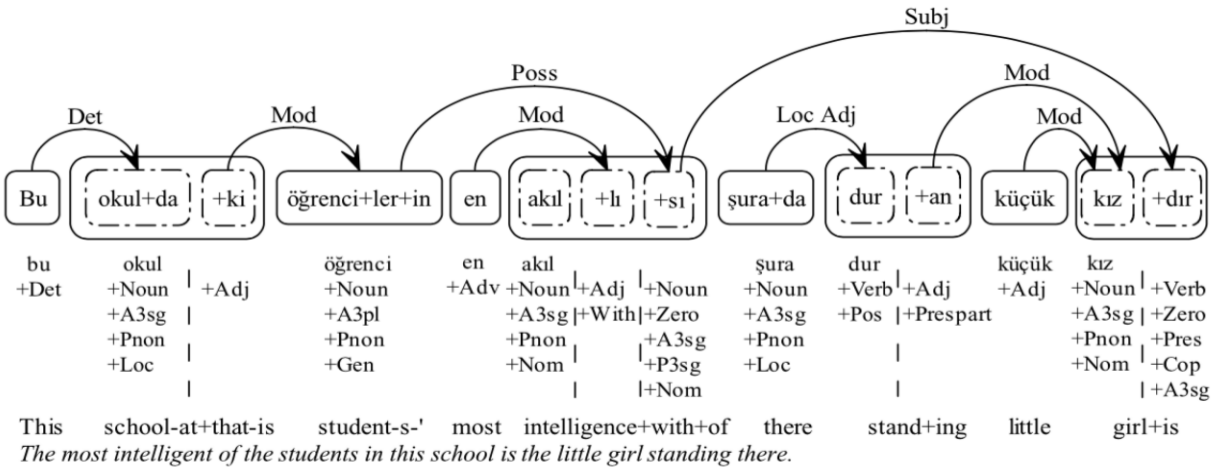

Figure 3. Inflection Groups (IGs) used in dependency parsing [13]

Oflazer (2014) [15] analyzes different NLP tasks on Turkish. In the dependency parsing task, the author underlines the importance of IGs and morphological units in dependency parsing.

Eryigit (2012) [16] makes an analysis on parsing in raw datasets in Turkish and shows that the locations of words in a sentence plays a crucial role in parsing.

Constituency Parsing

The constituency parse tree is based on the formalism of context-free grammars. In this type of tree, the sentence is divided into constituents, that is, sub-phrases that belong to a specific category in the grammar. In English, for example, the phrases "a dog", "a computer on the table" and "the nice sunset" are all noun phrases, while "eat a pizza" and "go to the beach" are verb phrases. The grammar provides a specification of how to build valid sentences, using a set of rules. As an example, the rule VP–>V NP means that we can form a verb phrase (VP) using a verb (V) and then a noun phrase (NP). While we can use these rules to generate valid sentences, we can also apply them the other way around, in order to extract the syntactical structure of a given sentence according to the grammar. Let's dive straight into an example of a constituency parse tree for the simple sentence, "I saw a fox":[17]

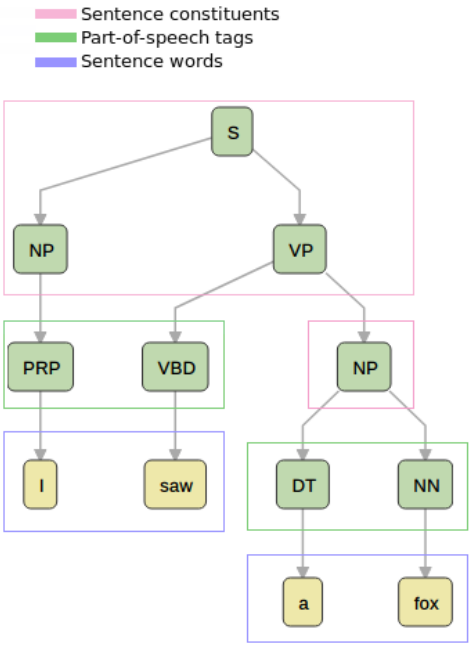

Figure 4. Constituency parse tree for the simple sentence [17]

Context free Chomsky grammar (CFG) is the most widely used formal system for modeling constituent structure in natural languages. CFG consists of a set of rules or productions, each of which expresses the ways that symbols of the language can be grouped and ordered together, and a lexicon of words and symbols. CFG G is defined by four parameters [2]:

$G = < Ns, S, Ts, R >$,

where Ns – a set of nonterminal symbols; $S \in N$ – a start nonterminal symbol; Ts – a set of terminal symbols; R – a set of rules of the form $A \rightarrow \alpha$, $A \in N$ – a nonterminal symbol, $\alpha \in (Ns \cup Ts,)*$ – a string of of symbols from the infinite set of strings ($Ns \cup Ts,)$*.

Constituency parsing and Dependency parsing

Constituency parsing focuses on identifying the constituent structure of a sentence, such as noun phrases and verb phrases.	Dependency parsing focuses on identifying the grammatical relationships between words in a sentence, such as subject-verb relationships.
Constituency parsing uses phrase structure grammar, such as context-free grammar or dependency grammar.	Dependency parsing uses dependency grammar, which represents the relationships between words as labeled directed arcs.
Constituency parsing is based on a top-down approach, where the parse tree is built from the root node down to the leaves.	Dependency parsing is based on a bottom-up approach, where the parse tree is built from the leaves up to the root.
Constituency parsing represents a sentence as a tree structure with non-overlapping constituents.	Dependency parsing represents a sentence as a directed graph, where words are represented as nodes and grammatical relationships are represented as edges.
Constituency parsing is more suitable for natural language understanding tasks.	Dependency parsing is more suitable for natural language generation tasks and dependency-based machine learning models.
Constituency parsing is more expressive and captures more syntactic information, but can be more complex to compute and interpret.	Dependency parsing is simpler and more efficient, but may not capture as much syntactic information as constituency parsing.

Constituency parsing is more appropriate for languages with rich morphology such as agglutinative languages.	Dependency parsing is more appropriate for languages with less morphological inflection like English and Chinese.
Constituency parsing is used for more traditional NLP tasks like Named Entity Recognition, Text classification, and Sentiment analysis.	Dependency parsing is used for more advanced NLP tasks like Machine Translation, Language Modeling, and Text summarization.
Constituency parsing is more suitable for languages with rich syntactic structures.	Dependency parsing is more suitable for languages with less complex syntactic structures.

Figure 5. Constituency parsing and Dependency parsing differences [20]

Conclusion

Uzbek is a Turkic language spoken by Uzbeks. Uzbek is spoken as either native or second language by 44 million people around the world making it the second-most widely spoken Turkic language after Turkish.[19] As an agglutinative language, Uzbek makes excessive use of morphological concatenation. With respect to syntactic properties, Uzbek has a relatively free word order. Even though SOV is the base word order, other permutations are highly utilized. For example:

Men maktabga onam bilan kecha bordim.
Kecha men onam bilan maktabga bordim
Men onam bilan kecha maktabga bordim
Maktabga kecha men onam bilan bordim

Uzbek has a free-order grammar and rich morphology, which makes dependency parsing even harder for Uzbek language. But, constituency parsing is more appropriate for languages with rich morphology such as agglutinative languages such as Uzbek.

References

1. Jurafsky, Dan; Martin, James H. (2021). *Speech and Language Processing*
2. Chomsky, N. 2002. Syntactic Structures. Mouton de Gruyter.
3. Ersin Ihsan Unkar (2007). PARSING TURKISH SENTENCES FOR NATURAL LANGUAGE WATERMARKING

4. Xavier Carreras and Michael Collins. Non-projective parsing for statistical machine translation. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 200–209. Association for Computational Linguistics, Singapore, 2009.

5. Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. Improved neural machine translation with a syntax-aware encoder and decoder. pages 1936–1945. 2017. P17-1177.

6. Katrin Fundel-Clemens, Robert Kuffner, and Ralf Zimmer. Relex – relation extraction using dependency parse trees. Bioinformatics (Oxford, England), 23:365–71, 2007.

7. Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2205–2215. Association for Computational Linguistics, Brussels, Belgium, 2018. D18-1244.

8. Zhanming Jie, Aldrian Obaja Muis, and Wei Lu. Efficient dependency-guided named entity recognition. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI’17, P3457–3465. AAAI Press, 2017.

9. Jenny Rose Finkel and Christopher D. Manning. Joint parsing and named entity recognition. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 326–334. Association for Computational Linguistics,Boulder, Colorado, 2009.

10. Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 344–354. Association for Computational Linguistics, Beijing, China, 2015. P15-1034.

11. Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph lstms. Transactions of the Association for Computational Linguistics, 5, 2017.

12. Salih Tuch. NEURAL DEPENDENCY PARSING FOR TURKISH. 2020. P. – 66.

13. Gulshen Eryigit, Joakim Nivre, and Kemal Oflazer. Dependency parsing of turkish. Computational Linguistics, 34(3):357–389, 2008.

14. Gulsen Eryigit and Kemal Oflazer. Statistical dependency parsing of turkish. 2006.

15. Kemal Oflazer. Turkish and its challenges for language processing. Lang.

Resour. Eval., 48(4):639–653, 2014.

16. Gulshen Eryigit. The impact of automatic morphological analysis & disambiguation on dependency parsing of Turkish. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12), pages 1960–1965. European Language Resources Association (ELRA), Istanbul, Turkey, 2012.

17. B.Elov, Sh.Hamraeva, D.Elova. Morfologik analizatorni yaratish usullari. O‘zbekiston: til va madaniyat (Amaliy filologiya), 2022, 5(1). 67-87-b.

18. O.Abdullayeva. Til korpuslarida lingvistik annotatsiya va uning prinsiplari. O‘zbek tilining milliy korpusi: muammo va vazifalar mavzusidagi xalqaro ilmiy-amaliy anjumani materiallari. – Toshkent, 31-may 2022-yil. – B. 130-136.

19. O.Abdullayeva. O‘zbek tili korpusida matnlarni sintaktik annotatsiyalash masalasi. // O‘zbek amaliy filologiyasi istiqbollari respublika ilmiy-amaliy anjumani materiallari. – Toshkent, 26-oktabr 2022-yil. – B. 122-126.

20. https://www.baeldung.com/cs/constituency-vs-dependency-parsing

21. https://en.wikipedia.org/wiki/Syntactic_parsing_(computational_linguistics)#CITEREFJurafskyMartin2021

22. https://en.wikipedia.org/wiki/Uzbek_language

23. https://www.geeksforgeeks.org/constituency-parsing-and-dependency-parsing/