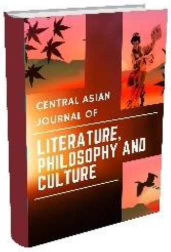


CENTRAL ASIAN JOURNAL OF LITERATURE, PHILOSOPHY AND CULTURE

eISSN: 2660-6828 | Volume: 04 Issue: 06 June 2023
https://cajlpc.centralasianstudies.org

Algorithm of Word Order and Syntactic Analysis in Uzbek Language Sentences

Hamroyeva Shahlo Mirdjanovna
Doctor of philological sciences, associate professor, Alisher Navo'i Tashkent State University of Uzbek Language and Literature

Abdullayeva Oqila Xolmo'minovna
Doctor of philosophy in philology, Senior teacher, Alisher Navo'i Tashkent State University of Uzbek Language and Literature

Rakhmonova Mokhira Abdusattor
TSULL Department of Computer Linguistics, 2nd year graduate student

Nizomova Fotima Bo'rikho'ja
TSULL Department of Computer Linguistics, 2nd year student

Received 4th Apr 2023, Accepted 5th May 2023, Online 14th June 2023

ANNOTATION

In this article, the word order and the place of sentence fragments in Uzbek language sentences, determining the correct order of sentences, the algorithm of syntactic analysis (parsing) of Uzbek language texts and the treebank structure are analyzed. Dependency parsing and Constituency parsing, which are syntactic analysis methods, are described. Simple sentence models, POS tagging, and part-of-speech tagging are shown. The Context-free grammar calculated by the syntactic parsing algorithm and graphs based on this grammar are given.

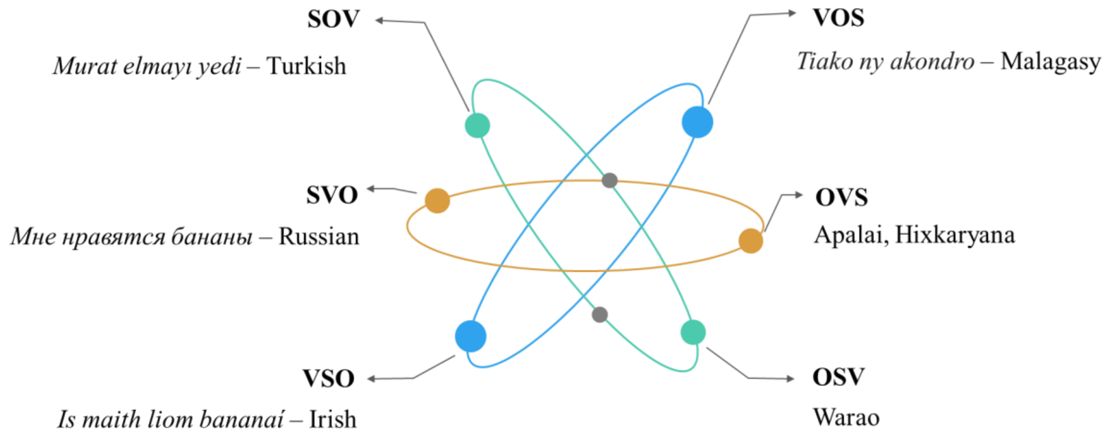
KEYWORDS: Word order, parsing, treebank, Subordinate analysis, Cluster analysis, CFG, Noun phrase, Verb phrase, Change-based approaches, Graph-based approaches.

Introduction. Word order – the arrengment of sentence components in accordance with certain grammatical rules, syntactic, semantic, stylistic aspects[https://uz.wikipedia.org/wiki/So'z_tartibi].

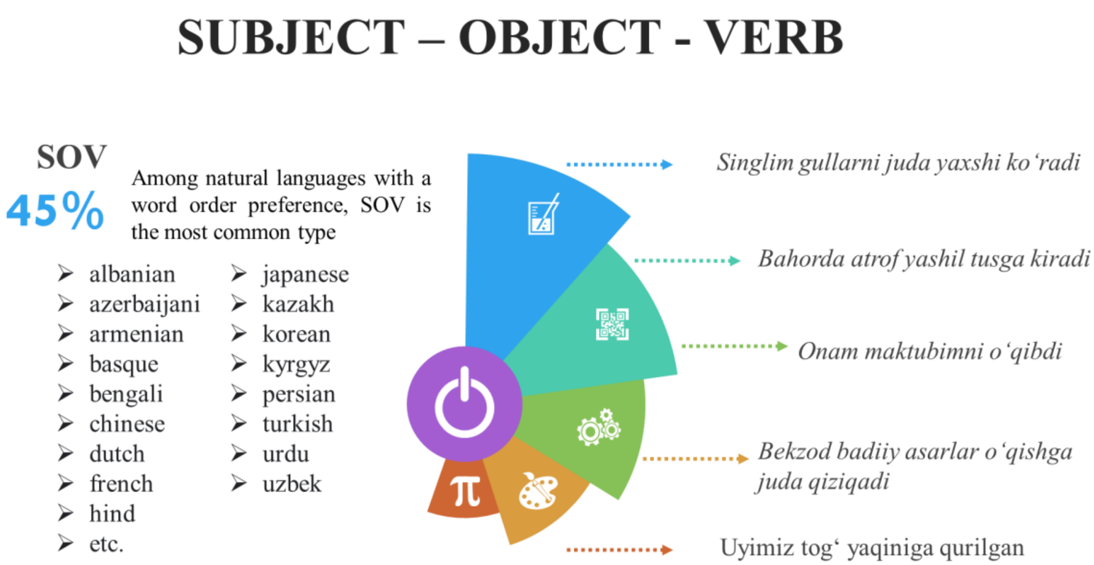
Starting from the second half of the 20th century, the American linguist Greenberg, based on the study of many languages, came to the conclusion that there are six fundamental orders of sentence components, including SOV, SVO, VSO, VOS, OVS, OSV, where S is the subject, O is the object and V is the Verb[Журинская М. А. 1990, 511-512].

CENTRAL ASIAN JOURNAL OF LITERATURE, PHILOSOPHY AND CULTURE

Volume: 04 Issue: 06 | June 2023, ISSN: 2660-6828



In Uzbek language, sentences are structured based on the SOV order.



The word order is a syntactic phenomenon that indicates the arrangement of sentence components in speech. In Uzbek language, the word order is mainly free, but it still has a dependent word order. This shows the unique characteristic of Uzbek language sentence structure. Some components may come in various places in sentence, while others come only in specific positions. When the syntactic position and function of words are determined by specific grammatical tools (such as classifiers, prepositions, auxiliary verbs, etc.), the word order is free. However, when determining the word order based on its position in a sentence, the word order becomes dependent. The word order affects both the syntactic and stylistic functions in the sentences. In Uzbek language, word order can serve as one of the grammatical tools that determines the syntactic relationship between words. For example: *Chaqqon bolalar yugurishdi*. In the sentence, the word *chaqqon* is placed after the word *bolalar* in the word order. This change in word order affects both the grammatical relationship and the meaning: *Bolalar chaqqon yugurishdi.* As we can see, the word order in such cases serves the syntactic function. In cases of balanced sentences (where the first parts of sentences express equal concepts), the change in the position of the first part also changes their syntactic function (in this case, no building or suffix is used); *Toshkent – O'zbekistonning poytaxti; O'zbekistonning poytaxti – Toshkent.* The change in the position of components in a sentence that contains an identifier and an identified part created a two-part sentence with a subject and predicate. *Oppoq paxtazor! – Paxtazor oppoq.* These are dependent word orders. In free word order, the arrangement of words does not necessarily convey the grammatical state and the meaning of the sentences a whole but instead emphasizes the importance of a particular part of the sentence, contributing additional information, additional tone and stylistic purpose to the general idea[Abdurahmonov, Shoabdurahmonov, Hojiyev, 1976: 176].

In a sentence, the position of the subject and verb. In Uzbek language, the subject usually comes at the beginning of the sentence with its related words, while the verb comes at the end of the sentence with its related words. Therefore, the word order of the subject and verb is in the form of "subject+verb": *Gullar ochildi*. Any related words that come before the main parts of the sentence can be considered as the related words of the subject, while any related words that come after the main parts of the sentence can be considered as related word of verb: *Supada o'tirgan onam menga qarab kuldi.* In a sentence, the subject usually comes before the verb and the verb comes after the subject which is the normal order. Conversely, the inverse order, where the verb comes before the subject, is possible as well. Inversion of the subject and verb occur in the following cases:

1. In literarcy language, to enhance the impact and emotionality: *Xalqlarimiz do'stligi mangu yashasin va yashnasin!*

2. In a dialogue: *Hammani ovora qildi bu shumtakalar* («Mushtum»).

3. In the author's sentence used after the quotation: *Bu qog'ozlar taxminlarga asoslangani uchun hujjat bo'lolmaydi, – dedi Yagona, uning so'zini kesib* (I. Rahim).

4. In poetry, as a result of the requirements of "vazn", "qofiya" and "turoq": *Keldi bahor, gul bahor, Erib bitdi oppoq qor* (Sh.Sa'dulla) [Abdurahmonov, Shoabdurahmonov, Hojiyev, 1976: 176].

The position of the object (in Uzbek: to'ldiruvchi) in a sentence. The position of the object in a sentence is determined in relation to what is being objected. The order of the object and what is being objected is usually in the form of "*object+what is being objected*". If several objects are attached to one what is being objected, the object indicating the object is usually closer to what is being objected. Therefore, an object without any indicators is placed before the filler with the indicators in relation to what is being objected: *Men sizga qadimgi Buxoroning yodgorliklarini ko'rsataman.* If the object is instrumental, its position is determined with respect to the what is being objected and is placed after it: *Buni senga oldim.* In Uzbek language, the inversion of the object and what is being objected is also possible. In this case, the inversion occurs in the function of the what is being objected. The conditions for this inversion are similar to the conditions for the inversion of the subject and verb, and are related to the emotional, literary and speech requirements that follow the brief introduction made by the author: *Ko'rmayin bosdim tikonni, tortadirman jabrini* (Uyg'un) [Abdurahmonov, Shoabdurahmonov, Hojiyev, 1976: 181].

The position of the adverb(in Uzbek: Hol) in a sentence. It stands before the "hollanmish"(adverb is the dependent of hollanmish). The positions of adverb in the sentence that is dependent of the "hollanmish" are follow: The style represented by the adverb is located close to the “hollanmish”: *U eshikdan sevinib kirib keldi*. Quantity-level adverb stands close to the “hollanmish”:*Biz siznikiga koʻp borganmiz*. Time and place adverbs come together before “hollanmish”, usually at the beginning of the sentence. In this case, the time adverb is used first, and the place adverb is used afterwards. *Kecha Kievdan bir oʻrtogʻim telefon qildi*. Cause and purpose adverbs can appear in different places of the sentence before being used: *Kasal boʻlganim uchun ishga vaqtida yetib kela olmadim.* This usual correct order may vary as logical emphasis demands. A logical accent adverb is close to the “hollanmish”(verb). In this case, incomplete inversion occurs. Application of the adverb after the “hollanmish”(verb) causes a complete inversion. Full inversion is found in lively language, emotional speech, constructions with excerpted sentences, poetry: *Ayta bersam shuki, bu oʻrtoq... qovunga piyoz qoʻshib yeganlar azbaroyi oʻrtoq Nurmatovni kuldirish uchun* (A.Qahhor) [Abdurahmonov, Shoabdurahmonov, Hojiyev, 1976: 182].

The position of the adjective (in Uzbek: Aniqlovchi) in a sentence. In the Uzbek language, the usual, correct order of “aniqlanmish” (adjective is dependent of aniqlanmish) with a adjective is in the form “adjective + aniqlanmish”. This order is fixed for qualifying with an adjective, and can be changed when viewed with an object, and sometimes interpreted with an explicative. An adjective with an “aniqlanmish”is in the correct order in prosaic speech, but in poetic speech it is sometimes inverted: *Ona sevmas farzand topilmas, Farzand yoʻqdir onani sevmas* (H.Olimjon). We have already said that the transformation of the adjective-“aniqlanmish” type of connection into the noun-verb type is not considered inversion. *Bayroq, havorang, magʻrur hilpirab turar* (H.Gʻulom) even in constructions of the type, the change of the order of the adjective and “aniqlanmish” cannot be considered as an inversion event. Because in inversion, the grammatical and semantic relations of the parts of the compound, their functions in the sentence should not change. *Havorang bayroq magʻrur hilpirab turar* gapida *havorang bayroq* if the combination comes into contact by association and forms an attributive relationship, *Bayroq, havorang, magʻrur hilpirab turar* in the sentence, it forms a semi-predicative relationship by entering into communication through interpretation. In the next case *havorang* word is considered a part of a complex sentence construction - a separate part. The usual correct order of reference with a referent is the form "referent-relative": *mening ukam*. This order can change depending on emotion in live speech, and depending on things like weight, rhyme, and stop in poetic speech, resulting in focus and subject inversion: Soʻzi qursin oʻshaning (Uygʻun) [Abdurahmonov, Shoabdurahmonov, Hojiyev, 1976: 183].

Syntactic analysis methods.

Syntactic parsing is the automatic analysis of syntactic structure of natural language, especially syntactic relations (in dependency grammar) and labelling spans of constituents (in constituency grammar) [Jurafsky, Dan, Martin, James, 2000: 927]. 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 [https://en.wikipedia.org].

Syntactic parsing is generally organized in two main parsing methods:

1. Dependency Parsing

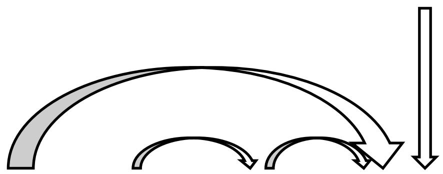
2. Phrase Structure Parsing [Unkar, 2007: 154]

CENTRAL ASIAN JOURNAL OF LITERATURE, PHILOSOPHY AND CULTURE

Volume: 04 Issue: 06 | June 2023, ISSN: 2660-6828

Dependency Parsing is the task of finding the grammatical structure of a sentence by identifying the syntactic and semantic relationships between words [Tunch, 2020: 1]. Dependency parsing has been utilized in many other NLP tasks such as machine translation, relation extraction, named entity recognition [14, 15], information extraction, 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. An example dependency graph for the sentence: *Men yaqin do'stimni ko'rdim.*

Root



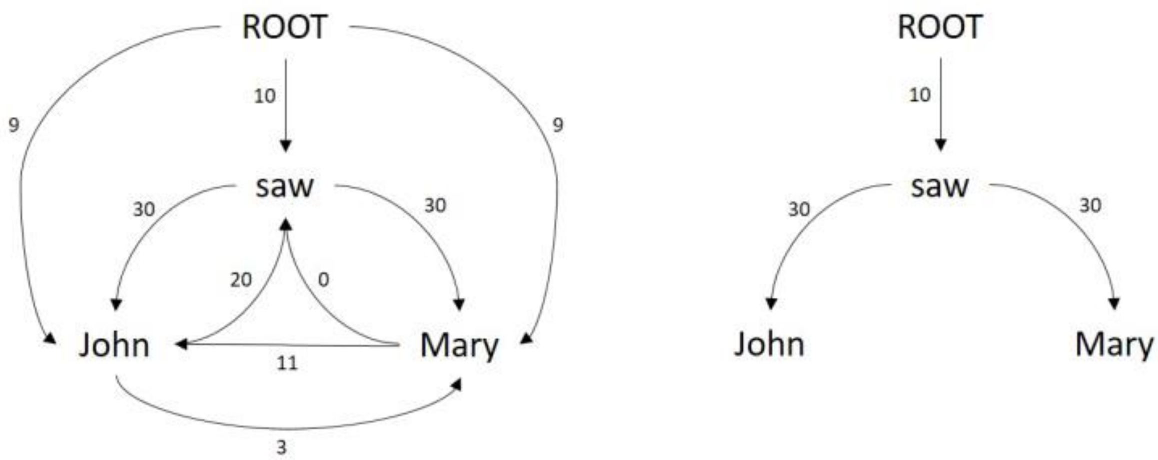
Men yaqin do'stimni ko'rdim

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 Men → ko'rdim; ”ko'rdim becomes the head and ”men” 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. There are two main approaches applied to dependency parsing problem in the literature:

1. Transition-based.
2. Graph-based. [Tunch, 2020: 30].

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 isthe root of the dependency tree; starting point of the whole dependency parsing process. [Tunch, 2020: 6].

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 6 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.

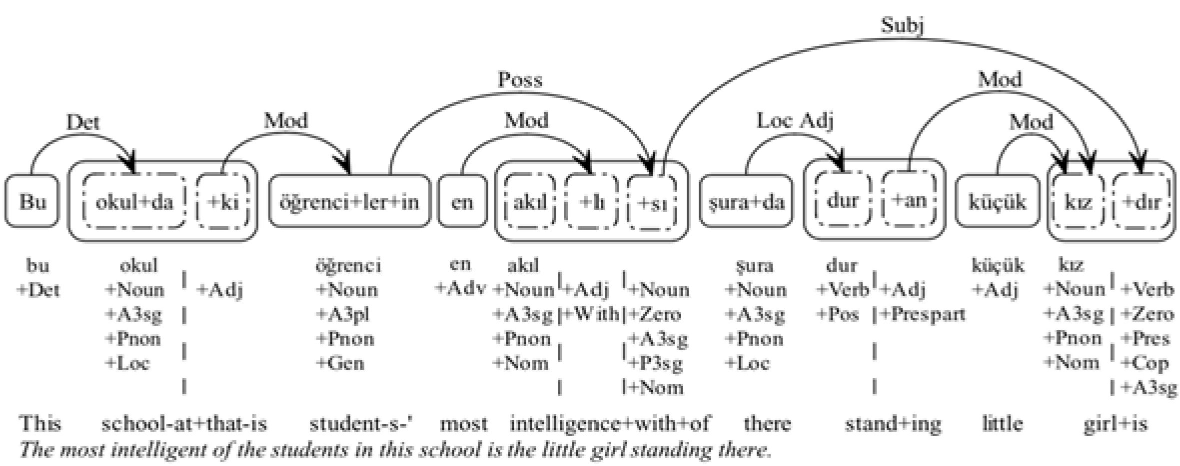


Graph-based Dependency Parsing [Tunch, 2020: 7]

Related Work on Dependency Parsing

Eryigit va Oflazer [Eryigit, Oflazer, 2006] 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 strictoutputs. As expected, IG-Based models give the best results[Tunch, 2020: 18].

Eryigit, Oflazer and Nivre [Eryigit, Nivre, Oflazer, 2008: 357–389] 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.



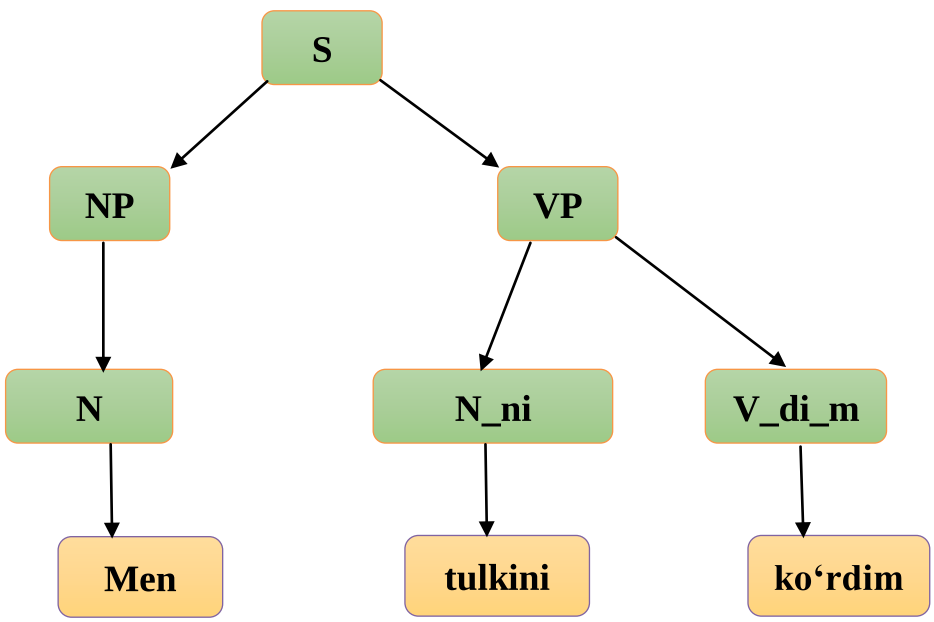
Inflection Groups (IGs) used in dependency parsing [Eryigit, Nivre, Oflazer, 2008: 357–389]

Oflazer [Oflazer, 2014: 639–653] analyzes different NLP tasks on Turkish. In the dependency parsing task, the author underlines the importance of IGs and morphological units in dependency parsing[Tunch, 2020: 19].

Eryigit [Eryigit, 2012: 1960–1965] 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[Tunch, 2020: 19].

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. [www.baeldung.com]. In Eglish, for example "qiziqarli kitob", "chiroyli kiyingan qiz" and "ajoyib kun botishi" are all noun phrases (NP), "pitsa yemoq" and "sohilga bormoq" are verb phrases (VP). The grammar provides a specification of how to build valid sentences, using a set of rules. As an example, the rule 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, "Men tulkini ko'rdim" degan oddiy gap uchun guruhlarga ajratib tahlil qilish daraxtiga to'g'ridan-to'g'ri misol keltiraylik [www.baeldung.com]:



Constituency parsing tree for simple sentence in Uzbek

Constituency Parsing and Dependency parsing [www.geeksforgeeks.org]

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	Dependency parsing is based on a bottom-up approach, where the parse tree is built from the leaves up to the

root node down to the leaves.	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.

Syntactic Parsing algorithms

Above, It is given that there are 2 main Parsing types:

1. Constituency parsing
2. Dependency parsing

Below is the algorithm of these 2 different parsing methods:

Constituency parsing

a) CKY (Cocke-Kasami-Younger) algorithm

b) Transition-based

c) Sequence-to-sequence

Dependency parsing

a) Transition-based

b) Grammar-based

c) Graph-based

In computer science, the **Cocke–Younger–Kasami algorithm** (alternatively called **CYK**, or **CKY**) is a parsing algorithm for context-free grammars published by Itiroo Sakai in 1961. [Sakai, Itiroo, 1961: 593–608]. The algorithm is named after some of its rediscoverers: John Cocke, Daniel Younger, Tadao Kasami, and Jacob T. Schwartz. It employs bottom-up parsing and dynamic programming. [en.wikipedia.org].

The standard version of CYK operates only on context-free grammars given in Chomsky normal form (CNF). However any context-free grammar may be algorithmically transformed into a CNF grammar expressing the same language [Sipser, Michael, 1997: 99].

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:

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 $A \to \alpha$, $A \in N$ – a nonterminal symbol, $\alpha \in (Ns \cup Ts,)$ * – a string of of symbols from the infinite set of strings $(Ns \cup Ts,)$* [Chomsky, 2002].

A CFG is a type of formal grammar that describes a language as a set of production rules that generate all possible combinations of syntax structures. These structures can then be used to model sentences and their associated meanings.

In natural language processing, CFG is used to create parsers (Parsers) that can accurately analyze and understand sentence structures. By specifying a set of word order rules, CFG helps identify the different parts of speech, phrases, and clauses contained in a sentence.

For example, the CFG rule "S -> NP VP" can be interpreted as "sentence - (S) consists of a noun phrase - (NP) followed by a verb phrase - (VP)". In general, CFG grammar is an important tool for sentence modeling and natural language processing because it can provide a framework for describing and understanding complex natural language structures.

Eshref Adali and Ilknur Dönmez, in their study the most appropriate formal grammar representing Turkish language is investigated. Accuracy of the suggested grammars' rules is evaluated in two different corpus. This study is the enhanced version of "Turkish Context Free Grammar Rules with Case Suffix and Phrase Relation" that was presented on UBMK 2016 International Conference on Computer Science & Engineering [Dönmez, Adalı, 2017: 33-40].

Altynbek Sharipbay, Banu Yergesh, Bibigul Razakhova, Gaziza Yelibaye and Assel Mukanova proposes a syntactic analysis of Kazakh simple sentences taking into account their semantics. To do this, first, the syntactic rules of sentences are described using formal grammar, then parsing trees and ontological models are built to determine the semantics of their components and the relationships between them. As a formal grammar used Chomsky's context-free grammar, and ontological models were built in the environment of Protége. [Sharipbay, Mukanova,Yergesh, Razakhova, Yelibayeva, 2019].

In this research work, to create a syntactic parsing model for simple sentences in the Uzbek language, first of all, models for all combinations of simple sentences are developed. Then, based on the developed simple sentence models, Context-free grammar is used, and formal grammar rules for simple sentences that can be understood by computer programs are drawn up.

CENTRAL ASIAN JOURNAL OF LITERATURE, PHILOSOPHY AND CULTURE

Volume: 04 Issue: 06 | June 2023, ISSN: 2660-6828

No	Simple sentences	POS tegging in Uzbek	POS tegging	The syntactic model of the sentence
1	Poyezd keldi	N+VB_di	N+V	E+K
2	Kitobni o‘qidi	N_ni+VB_di	N+V	T +K
3	Dalaga boryapman	N_ga+VB_yap man	N+V	H+K
4	Men taqdimga kechikdim	P+N_ga+VB_di m	PRON+N +V	E+H+K
5	Ali kimyo kitobini o‘qiyapti	N+N+N+ni+VB _yapti	N+N+N+V	E+A+T+K
6	Ali qo‘shiqni kuyladi	N+N_ni+VB_di	N+N+V	E+T+K
7	Ali bu yomon qo‘shiqni kuyladi	N+P+JJ+N_ni+ VB_di	N+PRON+ADJ+N+V	E+A+A+T+K
8	Anna kecha omadli edi	N+RR+N_li+V B	N+ADV+N+V	E+H+K
9	Oq bulutlar bilan qoplangan	JJ+N_lar+II+V B_n_gan	Adj+N+Aux+V	A+T+K
10	Sabrina kecha to‘pni dribling qildi	N+RR+N_ni+N +VB_di	N+ADV+N+N+V	E+H+T+K

Our next task is to make the simple sentence models compiled in the above table into a formal grammar in Context-free grammar. Noun phrase comes in the form of -NP and Verb phrase - VP.

Sentence: Poyezd keldi

CFG: S–> NP VP

VP–>V_di

NP–>N

N–>Poyezd

V_di–>keldi

Sentence: Dalaga boryapman

CFG: S–>VP

VP–> N_ga V_yap_man

N_ga–>Dalaga

V_yap_man–>boryapman

Sentence: Kitobni o‘qidi

CFG: S–>VP

VP–>N_ni V_di

N_ni–>Kitobni

V_di–>o‘qidi

Based on the CFG grammar, an algorithm for the correct order of simple sentences is created.

Conclusion

The complexity of sentence construction in Uzbek creates problematic situations in the process of automatic syntactic analysis. Studying the order of occurrence of fragments in sentences, identifying exceptional cases and developing a model of sentences based on this, first of all, developing an algorithm for the correct order of simple sentences will serve the construction of a syntactic analyzer of the Uzbek language in the future, developing parser and treebank programs. has a significant effect on output.

References

1. Altynbek Sharipbay, Assel Mukanova, Banu Yergesh, Bibigul Razakhova, Gaziza Yelibayeva. Syntax parsing model of Kazakh simple sentences, 2019.
2. Chomsky, N. Syntactic Structures, – Mouton de Gruyter, 2002
3. Dönmez, ˙I., & Adalı, E. Türkçe Hal Ekleri ve Öbekleri Kapsayan Ba˘glamdan Ba˘gımsız Dil Temsili Kuralları. – Türkiye Bili¸sim Vakfı Bilgisayar Bilimleri ve Mühendisli˘gi Dergisi, 2017. – 10(1), 33-40.
4. Ersin Ihsan Unkar (2007). PARSING TURKISH SENTENCES FOR NATURAL LANGUAGE WATERMARKING. – Bo´gazici University, 2007. – 154 b.
5. Gulsen Eryigit and Kemal Oflazer. Statistical dependency parsing of Turkish, 2006.
6. Jurafsky, Dan, Martin, James H. Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. – New Jersey: PrenticeHall, 2000. – 927 b.
7. Gulshen Eryigit, Joakim Nivre, and Kemal Oflazer. Dependency parsing of turkish. Computational Linguistics, 2008. – P. 357 – 389.
8. Gulshen Eryigit. The impact of automatic morphological analysis & disambiguation on dependency parsing of Turkish, – Istanbul: European Language Resources Association, 2012. – P. – 1960–1965.
9. Kemal Oflazer. Turkish and its challenges for language processing. Lang. Resour. Eval., 48(4):639–653, 2014.
10. Salih Tunch. NEURAL DEPENDENCY PARSING FOR TURKISH, 2020. – P. 66
11. Sakai, Itiroo. Syntax in universal translation. – Teddington, England: International Conference on Machine Translation of Languages and Applied Language Analysis, 1961. – P. 593–608.
12. Sipser, Michael. Introduction to the Theory of Computation (1st ed.). – 1997. – p. 99.

13. Журинская М. А. Типологическая классификация языков // Лингвистический энциклопедический словарь. Гл. ред. В. Н. Ярцева. Советская энциклопедия, 1990. – P. 511–512.
14. Abdurahmonov G‘.A., Shoabdurahmonov Sh.Sh., Hojiyev A.P. O‘zbek tili grammatikasi II T. Sintaksis. – Toshkent, 1976. – B. 181.
15. https://en.wikipedia.org/wiki/Syntactic_parsing_(computational_linguistics)#CITEREFJurafskyMartin2021
16. https://www.baeldung.com/cs/constituency-vs-dependency-parsing
17. https://en.wikipedia.org/wiki/CYK_algorithm#cite_note-1
18. https://www.geeksforgeeks.org/constituency-parsing-and-dependency-parsing/