Syntax Parsing: Implementation using Grammar-Rules for English Language

Madhuri A. Tayal
Research Scholar, G. H. Raisoni College of Engineering, Nagpur.
Asst. Prof. SRCOEM, Nagpur, INDIA.
madhuri_kalpe@rediffmail.com

Dr. M. M. Raghuwanshi
Principal, Rajiv Gandhi College of Engineering and Research
Nagpur, INDIA.
m_raghuwanshi@rediffmail.com

Dr. Latesh Malik
HOD(CSE),
G. H. Raisoni College of Engineering, Nagpur, INDIA.
latesh.malik@raisoni.net

***Abstract*--Syntactic parsing deals with syntactic structure of a sentence. The word 'syntax' refers to the grammatical arrangement of words in a sentence and their relationship with each other. The objective of syntactic analysis is to find syntactic structure of a sentence which is usually depicted as a tree. Identifying the syntactic structure is useful in determining the meaning of a sentence. Natural language processing is an arena of computer science and linguistics, concerned with the dealings amongst computers and human languages. It processes the data through lexical analysis, Syntax analysis, Semantic analysis, Discourse processing, Pragmatic analysis. This paper gives various parsing methods. The algorithm in this paper splits the English sentences into parts using POS tagger, It identifies the type of sentence (Facts, active, passive etc.) and then parses these sentences using grammar rules of Natural language. The algorithm has been tested on real sentences of English and it accomplished an accuracy of 81%.**

***Keywords:* Natural Language, sentences, phrases, grammar.**

I. Introduction

Language is the prime means of communication used by the individuals. It is the tool everyone uses to express the greater part of ideas and emotions. It shapes thought, has a structure, and carries meaning. Natural language processing is concerned with the progress of computational models of human language processing.

Syntax analysis is a fundamental area of research in computational linguistics. Semantic analysis is used in key areas of computational linguistics such as machine translation, storytelling, question-answering, information retrieval and information extraction [8], [15], [16].

Identifying the syntactic structure is useful in determining the meaning of the sentence. The identification is done using a procedure known as parsing. Syntactic parsing deals with the syntactic structure of a sentence. In many languages, words are brought together to form larger groups termed constituents or phrases, which can be modeled using context free grammar. Context free grammar is a set of rules or productions that expresses which elements can occur in a phrase and in what order.

Researchers have proposed a number of parsing methods for natural language sentences. [5], [6], [11], [12]. The main purpose of Syntax analyzer is to identify the syntactic structure of a sentence and parsing them accordingly. A widely used mathematical system for modeling constituent structure in natural language is context-free grammar (CFG) also known as phrase structure grammar[3]. Parse trees are used to show the structure of the sentence, but they often contain redundant information due to implicit definitions. The procedure to solve this problem is as follows. Initially it identifies the type of sentence like, Active sentences, Passive sentences, simple sentences etc. Then various components of these sentences are identified. The rearrangement amongst them is checked by the grammar rules given for every component of sentence. If the sentence parses through this grammar rules, then the sentence is syntactically correct. Otherwise it is syntactically incorrect. The detailed procedure is prescribed in various sections as follows. Section-2 provides the overview of CFG also known as phrase structure grammar.Section-3 represents various parsing approaches and procedures to solve this issue. Section-4 describes proposed method for checking syntax of the given sentences. Section-5 describes the experimental outcomes for the same and paper ends with conclusion and future work.

II. Context Free Grammar

Context-free grammar (CFG) was first defined for natural language by Chomsky (1957)[4] and used for the Algol programming language by [10]. A CFG consists of four components:

1. A set of non-terminal symbols, N
2. A set of terminal symbols, T
3. A designated start symbol, S, that is one of the symbols from N.
4. A set of productions, P, of the form:

$$A \dashrightarrow \alpha$$

Where A € N and α is a string consisting of terminal and non-terminal symbols. The rule A --› α says that constituent A can be rewritten as α. The simplified view of the grammar rules discussed so far is summarized.

S = NP VP
S = NPP VP
S = VP
S = NP NPP VP
S = NPP NPP NP VP

978-1-4799-2102-7/14 $31.00 © 2014 IEEE
DOI 10.1109/ICESC.2014.71

They are still more modified in the Section-4. The various abbreviations used for this grammar in given approach is mentioned in table-1 underneath.

TABLE I. LIST OF ABBREVIATIONS FOR THE GRAMMAR

Abbreviations	Abbreviations Meaning
S	Sentence
Det	Determiner
Adj	Adjective
Pron	Pronoun
Num	Numerals
Conj	Conjunction
Neg	Negation
Prep	Preposition
Adv	Adverb
V	Verb
VC	Verb Command
N	Noun
NP	Noun Phrase
VP	Verb Phrase
AP	Adjective Phrase
NPP	Noun Preposition Phrase
VPP	Verb Preposition Phrase
APP	Adjective Preposition Phrase

III. Parsing approaches

A CFG defines the syntax of a language but does not specify how structures are assigned. The-task that uses the rewrite rules of a grammar to either generate a particular sequence of words or reconstruct its derivation (or phrase structure tree) is termed parsing [7]. A phrase structure tree constructed from a sentence is called a parse.

A. Top down parsing

Top down parsing starts its search from the root node S and works downwards towards the leaves. The fundamental assumption here is that the input can be derived from the chosen start symbol s, of the grammar. The next step is to find all sub-trees which can start with s. To generate the sub trees of the second –level search, we expand and root node using all the grammar rules with s on their left hand side. Likewise, each non-terminal symbol in the resulting sub-trees is expanded next using the grammar rules having a matching non-terminal symbol on their left hand side. The right hand side of grammar rules provides the nodes to be generated, which are the expanded recursively. As the expansion continues, the tree grows downward and eventually reaches a state where the bottom of the tree consists only of part-of-speech categories. At this point, all trees whose leaves do not match words in the input sentence are rejected, leaving only trees that represent successful parses.

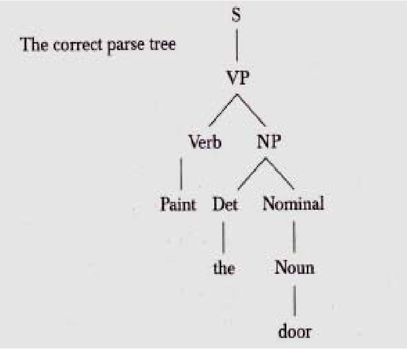

Figure 1. A top-down search space

B. Bottom-up Parsing

A bottom-up parser starts with the words in the input sentence and attempts to construct a parse tree in an upward direction towards the root. At each step, the parser looks for rules in the grammar wh-ere the right hand side matches some of the production in the parse tree constructed so far, and reduces it using the left hand side of the production. The parse is considered successful if the parser reduces the tree to the start symbol of the grammar.

Each of these parsing approaches has its advantages and disadvantages. As the top-down search starts generating trees with the start symbol. The grammar, it never wastes time exploring a tree leading to a different root. However, it wastes considerable time exploring S trees that eventually result in words that are inconsistent with the input. This is because a top down parser generates trees before seeing the input. On the other hand, a bottom-up parser never explores a tree that does not match the input. However, it wastes time generating trees that have no chance of leading to an S-rooted tree.

Many attempts have been made to develop syntax parsing with various approaches [1][2][8][9]. Majority of approaches to check syntax correctness is based on probabilistic approach [6][13].The proposed approach for syntax analysis is specified in the next section.

IV. Algorithm (Rule based approach)

The task of syntax analyzer is done through following algorithm.

1. Enter a sentence.
2. Categorize the sentence using Table-3.
3. Check the phrases of sentences using various tags returned by POS tagger[14]. (Its noun phrases (N, NP, NPP) and (V, VP, VPP)).
4. Partition the sentence into NP and VP identified in Table-4.
5. Parse the NP, NPP, V and VPP by matching it against Grammar rules.
6. If all parts of the sentences are parsed correctly then sentence is syntactically correct, else the sentence is syntactically incorrect.

A. POS Tagger

A Part-of-Speech Tagger (POS Tagger)[14] is a portion of software that reads text in some language and allocates parts of speech(i.e. tags) to each word. It assigns a part-of-speech like noun, verb, pronoun, preposition, adverb, and adjective or other lexical class marker to each word in a sentence. This software is a Java implementation of the log-linear part-of-speech taggers. A number of of Taggers are available Stanford Tagger, Apache UIMA Tagger; Eric Brill's simple Rule Based Tagger etc. are some of them. Out of which Stanford tagger has been used. Its basic download contains two trained tagger models for English. The full download contains three trained English tagger models, an Arabic tagger model, a Chinese tagger model, and a German tagger model. Both versions include the same source and other required files. The tagger can be retrained on any language, given POS-annotated training text for the language. The input to a tagging algorithm is a string of words of a natural language sentence and a quantified tag set (a finite list of Part-of-speech tags). The output is a single finest POS tag for each term shown in table-2.

TABLE 2. POS TAGGED OUTPUT AND THEIR MEANINGS.

Tagger o/p	**Meaning**	**Tagger o/p**	**Meaning**	**Tagger o/p**	**Meaning**
CD	Cardinal Number	NNPS	Proper Noun, plural	TO	to
CC	Coordinating conjunction e.g. and, but, or...	NNS	Noun, plural	VBN	, past participle
DT	Determiner	PDT	Predeterminer e. g. all, both ... when they precede an article	UH	Interjection e.g. uh, well, yes, my...
EX	Existential there	POS	Possessive Ending e.g. Nouns ending in 's	VB	Verb, base form subsumes imperatives, infinitives and subjunctives
FW	Foreign Word	PRP	Personal Pronoun e.g. I, me, you, he...	VBD	Verb, past tense includes the conditional form of the verb to be
IN	Preposition or subordinating conjunction	PRP$	Possessive Pronoun e.g. my, your, mine, yours...	VBG	Verb, gerund or present participle
JJ	Adjective	RB	Adverb Most words that end in -ly as well as degree words like quite, too and very	VBP	Verb, non-3rd person singular present
JJR	Adjective, comparative	RBR	Adverb, comparative Adverbs	VBZ	Verb, 3rd person singular present
JJS	Adjective, superlative	RBS	Adverb, superlative	WDT	Wh-determiner e.g. which, and that when it is used as a relative pronoun
LS	List Item Marker	RP	Particle	WP	Wh-pronoun e.g. what, who, whom...
MD	Modal e.g. can, could, might, may...	SYM	Symbol used for mathematical, scientific symbols	WP$	Possessive wh-pronoun
NN	Noun, singular or mass	TO	to	WRB	Wh-adverb e.g. how, where why
NNP	Proper Noun, singular				

B. Categorization based on Kinds of Sentences and Grammar Rules

According to Wren and Martin [17] the sentence comprises of Subject, Verb and Object. So, each sentence has to have a subject(S), Object (O) and a Verb (V). Some sentences may have adjectives, adverbs and conjunctions. There are also sentences which are interrogative i.e. they ask a question. Keeping all these in mind, sentences are categorized in different type. It is important to categorize sentences because the POS tagger treats the sentences as group of words. It does not look at the meaning of the sentence as a whole. The basis for the process of categorization is shown in the table 3. The categorization is as follows:

1. Sentences having exactly one subject, one verb and one object. (Simple)
2. Sentences having exactly one subject, one verb, one object and adjectives also.(Simple with ADJECTIVES)
3. Sentences containing more than one noun and verbs. (COMPLEX)
4. Sentences contains question. (INTERROGATIVE)
5. Sentences containing conjunctions(CONJUCTIONS)
6. Simple fact statements. (FACTS)
7. Sentences in active form. (ACTIVE).
8. Sentences in passive form. (PASSIVE).

TABLE 3. CATEGORIZATION OF ENGLISH SENTENCES

Basis of categorization	Category
Sentence with only one subject, one verb and one object.	Simple
Sentence with only one subject, verb, and adjective followed by a verb.	SVO with adjective
Sentences with more than one subject or object and having "and"…"or" in it.	Complex
Sentences terminating with a "?".	Interrogative
Sentences containing conjunctions.	Conjunctions
Sentences starting with This, That.	Facts
Simple Sentences.	Active
Sentences in which the subject follows "by".	Passive

This categorization has been made to check for the accuracy of this system in respect to types of sentences. After categorizing the sentences the format of sentences using POS tagger is checked. POS tagger identifies the noun phrases (N, NP, NPP) and (V, VP, VPP) using the tags mentioned in the Table-2. Then partition the sentence into different phrases like NP and VP defined in Table-4. Then it Parses the NP, NPP, V and VPP by matching it against Grammar rules. Grammar rules (from Table-4) have been implemented for English language sentences and identified that they are working for different types of sentences (Simple, complex, active, passive etc.) using table-3. Table-4 shows the grammar rules to be checked for the syntax analyser.

TABLE-4. RULES FOR THE FORMATION AND CHECKING OF DIFFERENT PHRASES

Sr. No.	Phrases	Phrases and Rules
1.	S	i. S = NP VP ii. S = NPP VP iii. S = VP iv. S = NP NPP VP v. S = NPP NPP NP VP
2.	NP	i.NP = N ii. NP = Det Adj N iii. NP = Det N iv. NP = Pron v. NP = Pron N vi. NP = Num N vii. NP = Num N N viii. NP = N Conj N ix. NP = Num N N Conj N x. NP = Det N N xi. NP = Det Adj Adj N xii. NP = Pron N N xiii. NP = Adj Pron N xiv. NP = Det Adj N N xv. NP = Det Adj N Pron xvi. NP = Neg N xvii. NP = Pron Adj N
3.	NPP	NPP = Prep NP
4.	AP	i. AP = Adj ii. AP = Adj Adj iii. AP = Adj Conj Adj
5.	APP	APP = Prep AP
6.	V	i.V = V ii. V = V V iii. V = V Adv V iv. V = V Neg V v. V = V V V V vi. V = V Conj V vii. V = V Adv viii. V = V Neg V Adv ix. V = Adv Conj Adv x. V = Adv V Neg V xi. V = V Adv Conj Adv xii. V = Adv V xiii. V = V V Adv
7.	VPP	VPP = Prep V
8.	VP	i.VP = V NP ii. VP = V VPP NP iii. VP = V NPP NP iv. VP = V NP NPP v. VP = V AP vi. VP = V NP NP VPP vii. VP = V viii. VP = V NPP

		ix. VP = V VPP x. VP = V NP V xi. VP = V NP VPP NP xii. VP = V VPP NPP xiii. VP = V NP NPP V NP xiv. VP = V NP AP xv. VP = V NP AP VPP xvi. VP = V NPP NPP xvii. VP = V NP V NPP xviii. VP = V VPP NP NP xix. VP = V NP NPP NPP xx. VP = V NPP NPP NPP xxi. VP = V VPP AP NPP NPP xxii. VP = V VPP NP NPP xxiii. VP = V AP NPP NPP xxiv. VP = V NP AP NPP xxv. VP = V NPP AP xxvi. VP = V VPP NP AP xxvii. VP = V AP NPP xxviii. VP = V NP VPP NP NPP xxix. VP = V NP NPP xxx. VP = V NPP VPP NP xxxi. VP = V NPP AP NPP

The analysis of words in a sentence is to know the grammatical structure of the sentence. The words are converted into constructions that show how the words relate to each other. Some of the sentences may be prohibited if they disrupt the rules of the language for how words may be combined. Syntax accuracy has been verified for the sentences and their corresponding results are shown in next section.

V. RESULTS

For experimentation sample sets had chosen for different categories of sentences such as simple, complex, active, passive voice, questions etc., and each holds 50 random sentences. So, overall 400 samples have been verified. The algorithm has accomplished an accuracy of 81%. The sample sentences and their corresponding syntactic understanding whether they are syntactically correct or not shown in the table 5.

TABLE 5. RESULTS OF SYNTAX ANALYSER

Type of sentence	Sample Sentences	Output
Simple	1. The angry girl kicked the ball. 2. She went to school. 3. I want to know your name. 4. They lived in a huge palace.	Sentence is syntactically correct.
Simple+ADJ	1. Rahul is a clever boy. 2. He likes tasty pizza. 3. I love fresh flowers. 4. Jack likes to visit lovely places.	Sentence is syntactically correct.
Complex	1. They were having a good time. 2. They were playing in the ground 3. They were studying in the good college. 4. He was selling fruits in front of the hall.	Sentence is syntactically correct.
Questions	1. Who are you? 2. What is your name? 3. When is your birthday? 4. What is the name of your village?	Sentence is syntactically correct.
Conjunctions	1. He was put behind the bars for his crime. 2. The cat was sitting under the chair. 3. The children performed fabulously in the concert. 4. They went to the park and played football.	Sentence is syntactically correct.
Facts	1. Hellen Keller was blind. 2. Sun rises in the east. 3. The earth is round. 4. The universe is infinite.	Sentence is syntactically correct.
Active sentences	1. The girl was washing the car. 2. Sita writes a letter. 3. Rita wrote a letter. 4. Rahul has written a letter.	Sentence is syntactically correct.
Passive sentences	1. The car was being washed by the girl. 2. A letter is written by Sita. 3. A letter has been written by Teena.	Sentence is syntactically correct.
Incorrect sentences	1. Sita a letter. 2. Rita wrote a. 3. The girl washing the car. 4. Boy the go the to store	Sentence is syntactically incorrect.

CONCLUSION AND FUTURE WORK

This paper focuses on syntax analysis for natural language. It also describes the syntax Representation for English Language. This research paper presents an approach to check syntactic correctness of the sentence. This approach achieved an accuracy of 81%. The accuracy of the system can be further increased through corpus training. In future using some more techniques, performance of the system will be improved.

REFERENCES

[1] Bharti Akshar and Rajeev Sangal, “A Karaka-based approach to parsing of Indian languages”, Proceedings of the 13th Conference on Computational Linguistics, Association for Computational Linguistics.

[2] Bharti Akshar, Vineet Chaitanya, and Rajeev Sangal, Natural Language Processing: A Paninian Perspectiue, Prentice-Hall of India, 1995.

[3] Charniak, Eugene, Statistical Language Learning, MIT press, Cambridge, 1993.

[4] Chomsky, N., Mouton,“Syntactic Structures”, The Hague, 1957.

[5] Collins, MJ., “Head-driven statistical parsing for natural language processing,” Ph.D. Thesis, University of Pennsylvania, Philadelphia.

[6] Infante-Lopez, Gabriel and Maarten de Rijke, “A note on the expressive power of probabilistic context free grammars”, Journal of Logic, Language and Information, Kluwer Academic publisher, l5 (3), 2006.

[7] Jurafsky, Daniel and James H. Martin, “Speech and Language Processing:An Introduction to Natural Language Processing” Computational Linguistics, and Speech Recognition, Prentice Hall, NJ, , 2000.

[8] Manning, C. and H. Shutze, “Foundations of Statistical Natural Language Processing”, MIT Press, Cambridge, 1999.

[9] Marcus. Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz, “Building a large annotated corpus of English: the Penn treebank,'Computational Linguistics, 19, pp. 313-30, 1993.

[10] Naur, Peter,J.w. Backus , F.L. Bauer,J. Green , C.Katz,J. McCarthy, A.Perlis, H. Rutishauser, K. Samelson, B. vauquois, J. H. wegstein, A.van Wijngaarden, and M. Woodger, 'Report on the algorithmic language ALGOL 60,' communications of the ACM, 3(5), pp. 299-314, 1960.

[11] Ney, H., 'Dynamic programming parsing for context-free grammars in continuous speech recognition,' IEEE Transactions 0n Signal Processing 39(2), pp. 336-40, 1991.

[12] Backus, J.W., “The syntax and semantics of the proposed International algebraic language of the zurich ACM-GAMM conference”, Proceedings of the international Conference on Information Processing, UNESCO, pp.125-32, 1959.

[13] Claire M. Nelson, Rebecca E. Punch, John Donaldson, “An Interactive Software Tool for Parsing English Sentences”, Proceedings of the Midstates Conference on Undergraduate Research in Computer Science and Mathematics, 2011.

[14] STANFORD POS TAGGER: http://nlp.stanford.edu/software/tagger.shtml

[15] Rich and Knight, “Artificial Intelligence”, TATA Mc Graw Hill Second Edition.

[16] Tanveer Siddiqui, U.S. Tiwari, Natural language Processing and Information Retrieval, Oxford University Press.

[17] Wren and martin, English grammar and composition. S. Chand & Company LTD.