IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-4, Issue-3 E-ISSN: 2347-2693

English Grammar Checker

Pratik Ghosalkar[1*], Sarvesh Malagi[2], Vatsal Nagda[3], Yash Mehta[4], Pallavi Kulkarni[5]

[1,2,3,4,5]Computer Science Department,
K. J. Somaiya College of Engineering, India

www.ijcseonline.org

Received: Mar/02/2016 Revised: Mar/10/2016 Accepted: Mar/24/2016 Published: Mar/31/2016

Abstract— 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. The usually poor quality of grammar leaves a bad impression on the reader. Therefore, there is a need for grammar checkers. We propose a grammar checking system by means of 'Syntax Analysis'. [1][9]Syntax refers to the arrangement of words in a sentence and their relation with each other. The objective of syntactic analysis is to find syntactic structure of a grammar of a natural language. Natural language processing is an area of computer science and linguistics, concerned with the dealings amongst computers and human languages. It processes the data through lexical analysis, Syntax analysis and Semantic analysis. This paper gives various parsing methods. The algorithm specified in the paper splits the English sentences into parts using POS tagger and then parses these sentences using grammar rules of Natural language.

Keywords—Natural Language Processing, Context-Free-Grammar, CYK Algorithm, Part-of-Speech Tagging, Syntax Parsing

I. INTRODUCTION

The issue of grammar correctness is of great importance. A grammar checker application can be useful to evaluate a person's proficiency in English grammar. The purpose of this project arises out of the need for checking of grammar in text on the computer. This will help in recognizing the errors in the grammar of the given to a certain extent based on the rules defined. [1]Natural language processing is a field of artificial intelligence, and computational linguistics linked between computers and human languages. As such, natural language processing is re1lated to the area of human–computer interaction. Modern natural language processing algorithms are based on machine learning, especially stochastic machine learning. The concept of machine learning is different from that of most prior attempts at language processing. The field of Natural Language Processing (NLP) helps in analyzing and processing spoken language. Natural Language Processing deals with understanding natural languages i.e. high level languages. The scope of our project lies in the concept of part-of-speech tagging and parsing for grammar correction. We aim to check the correctness of the English grammar of the given input text. We aim to design a syntax analyzer which checks for syntactical error in a given sentence based upon the rules of grammar designed for English language, which will be implemented in the form of CFGs for parsing algorithm. Accurately designed CFGs with suitable algorithm for parsing is the core of our project.

This paper gives an overview about how to implement an English grammar checker using syntactic parsing and properly designed CFGs for English Language. Firstly, POS tagging is done, then parsing with designed CFGs and lastly error detection, if any.

II. SYSTEM OVERVIEW

The proposed system is a client – server based system. It is designed for a real-time grammar checking. It consists of four major components. These four components are functionally dependent on each other. The system follows a hybrid concept, basically to increase the accuracy of tagging input we use stochastic POS-Tagger and try to check the correctness of grammar using a rule based parser. The parser generates the parse tree, if the parser can build parse tree successfully then we assume that the entered input is error free, while if it fails to construct, then some post processing is done to the incorrectly constructed parse tree and error is detected and displayed. The system detects grammatical error in the sentences given to it.

The fig. 1.1 shows the brief description of the system and its components. The system consists of four major components and a web based user interface can be distributed over the internet or other network. Here client enters the input text, the real time 'sentencifier' or sentence detector will detect sentence and it will pass this sentence to a stochastic POS Tagger. In this system we propose the use of Stanford Maxent Part of speech tagger. The tagger tags the input suitable and it will generate a stream of tokens for parsing.

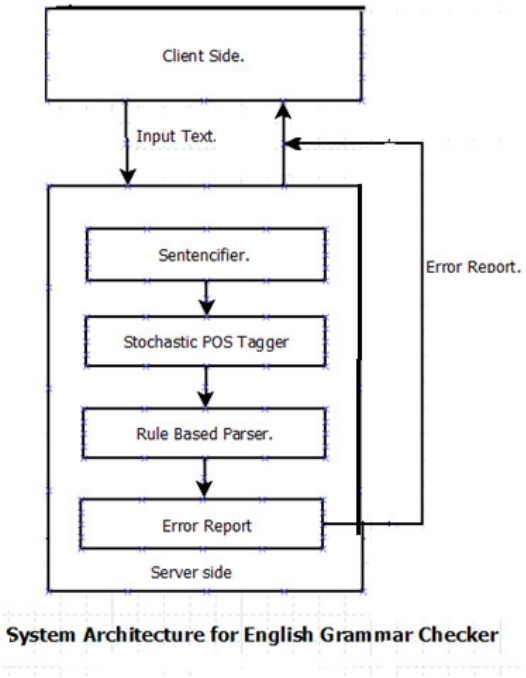

Fig. 1.1

Now, the stream of tokens is forwarded to a Rule based parser, which will do syntax analysis over the stream of tokens to generate parse tree or an error token. With the help of error token if generated any, the error report gives back error feedback to user.

A. Sentence Boundary Detection[3,4]

It is the component which will interact with user input, it will take input text to recognize in real time to extract a sentence. It is proposed to use Apache Open NLP project, which gives the functionality to determine valid sentence from input. It uses the concept of machine learning to improve its accuracy.

B. Part-of-Speech Tagging[3]

The POS-Tagger or part of speech tagger, tags the input words in a sentence with suitable tags. As mentioned earlier the POS-Tagger which can used is Maxent POS-Tagger by Stanford University. It is a stochastic based Tagger, which looks into probability distribution form given set of corpus. The corpus is collection of Wall Street Journal articles with each word tagged with a Part-Of-Speech. This gives output as the input sentence with POS tag for each word.

C. Syntactic parsing

Syntactic Parsing is basically core of the system, the syntax analysis on the sentence is the process of determining the relationship between tokens. If the relationship found between token agrees the permitted relationships governed by English grammar, here in this case CFGs. [9]The designed CFGs define the rules of English grammar which are used for validating the correctness of English grammar. It also performs error detection i.e. what part of sentence is incorrect.

For the given problem, two algorithms have been studied viz. Earley Parsing algorithm and Cocke-Younger-Kasami (CYK) algorithm. Brief explanation for both algorithms is given below.

i. Cocke-Younger-Kasami (CYK) algorithm:

[7]CYK algorithm is a parsing algorithm which parsers Context Free Grammars in Chomsky Normal Form (CNF). The grammar has to be in CNF form because the algorithm checks for the splitting possibility which is half in CNF. CYK algorithm is also called as table filling algorithm because it proceeds by filling a triangular table with suitable grammar entries and then working in bottom up fashion to fill the table. The bottom row has strings of length 1, second last row has strings of length 2 and so on. Compare at most n pairs of previously computed sets:

$(X_{i,\,i}, X_{i+1,\,j})$, $(X_{i,\,i+1}, X_{i+2,\,j})$ …… $(X_{i,\,j-1}, X_{j,\,j})$.

Then this is repeated till we reach the topmost row. If the topmost row has Start variable, from the designed CFGs, in it, then it indicates that the given sentence can be generated by the grammar, if the top row doesn't have the Start symbol then it indicates that the String cannot be generated by the given grammar[5,6]. To obtain a parse tree of the String generated we maintain a tree data structure which is linked to elements of the array. The worst case time complexity of CYK parsing algorithm is, where n is the length of the string that is parsed and |G|, the size of the grammar in CNF. This algorithm is amongst the best algorithms based on worst case running complexity. The time complexity of this algorithm is $O(n^3)$ for unambiguous grammars. It doesn't work with ambiguous grammars. Triangular table construction can be represented as given in fig. 1.2

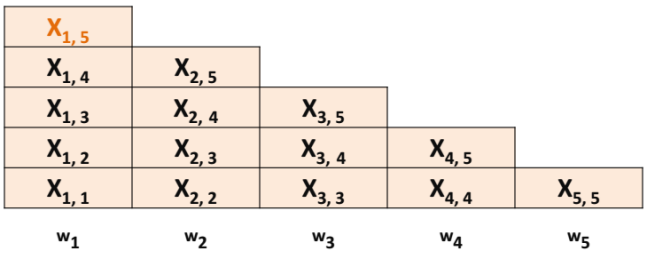

Table for string 'w' that has length 5

[7]Fig. 1.2 Constructing Triangular Table

ii. Earley Parsing Algorithm

[2][8]Earley parsing algorithm is a hybrid type of parsing algorithm. It combines the predictive rules of top-down approach with robust bottom-up parsing. The algorithm was designed for purpose of NLP. The algorithm maintains dotted rules which help to keep a track of part of input, which has already been seen before. Hence whenever any predictions go wrong, then it doesn't have start all over again. Foe time complexities, unambiguous grammars are $O(n^2)$ time complex and ambiguous grammars are $O(n^3)$ time complex.

For the system here it is decided to opt for CYK algorithm for its ease of use and in designing CFGs for English grammar

D. Error Reporting

The error report module is designed for mapping parsing errors to actual grammatical error. This component is designed using Web-based technologies, which will help to manipulate input text to display the incorrect part of sentence, if any, to the user. Basically it is intended to highlight error causing words. The reported errors will be displayed to the user on a web page.

III. CONCLUSION

In this paper, various parsing algorithms for natural language processing are studied and CYK algorithm has been chosen to be implemented. Sentence boundary detection will be done using NLTK tool Maxent tagger which contains tagged corpus will help in tagging the words in the sentences with POS tags. Context free grammars are to be designed for English grammar on which the CYK algorithm shall work. The accuracy of designed context free grammars will define the accuracy of the system as whole. The incorrect part from the sentence shall be detected using the algorithm and errors shall be marked, if any.

REFERENCES

[1] Earley Parser, https://en.wikipedia.org/wiki/Earley_parser, 27/11/2015
[2] The Earley Parsing Algorithm, http://demo.clab.cs.cmu.edu/fa2014-11711/images/a/a6/Earley-Parsing.pdf , 27/11/2015
[3] Kinoshita, J.; Salvador, L.N.; Menezes, C.E.D.; Silva, W.D.C., "CoGrOO - An OpenOffice Grammar Checker," in Intelligent Systems Design and Applications, 2007. ISDA 2007. Seventh International Conference on , vol., no., pp.525-530, 20-24Oct.2007 doi: 10.1109/ISDA.2007.145
[4] Jaiswal, U.C.; Kumar, R.; Chandra, S., "A Structure Based Computer Grammar to Understand Compound-Complex, Multiple-Compound and Multiple-Complex English Sentences," in Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT '09. International Conference on , vol., no., pp.746-751, 28-29Dec.2009 doi:10.1109/ACT.2009.189
[5] Lee, J.; Seneff, S., "An analysis of grammatical errors in non-native speech in English," in Spoken Language Technology Workshop, 2008. SLT 2008. IEEE , vol.,no.,pp.89-92,15-19Dec.2008 doi:10.1109/SLT.2008.4777847
[6] Brian Roark (Oregon Health & Science University), Kristy Hollingshead (University of Maryland), Nathan Bodenstab (Oregon Health & Science University), "Finite-State Chart Constraints for Reduced Complexity Context-Free Parsing Pipelines", in Journal: Computational Linguistics, Volume 38 Issue 4, December 2012, Pages 719-753 , doi:10.1162/COLI_a_00109
[7] The CYK Algorithm, https://www.cs.wmich.edu/~elise/courses/cs6800/CYK-Algorithm.ppt, 16/11/2015
[8] Earley parser.pdf - Computer Science and Engineering, www.cse.unt.edu/~tarau/teaching/NLP/Earley%20parser.pdf, 16/12/2015
[9] M. A. Tayal, M. M. Raghuwanshi and L. Malik, "Syntax Parsing: Implementation Using Grammar-Rules for English Language," Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on, Nagpur, 2014, pp. 376-381. doi: 10.1109/ICESC.2014.71