Grammar Checker

Nivedita Mandal
B.Tech C.S.E
Galgotias University
Greater Noida, India

Nikita
Associate Professor,
School of Computing Science and Engineering,
Galgotias University,
Greater Noida, Uttar Pradesh, India

Aditya Anand
B.Tech A.I.M.L
Galgotias University
Greater Noida, India

Yash Garg
B.Tech C.S.E
Galgotias University
Greater Noida, India

Abstract:- This paper models the working of Machine Learning algorithm to detect errors from a given sentence or a paragraph. When the user input the text then it highlight the error in the text. The database contains the correct words which we generally use in English language. The errors are checked in the database and then it returns the correct word for the highlighted error. It also highlight the corrected word in the corrected text so that the user will understand the mistakes. At last we get the corrected text.The types of error which can be in a sentence written in an English language are spelling, misused, split and merge, subject verb agreement, common and proper, double words, comparative superlative, indefinite article, definite article, plurality, pronouns, consecutive nouns and tenses.We developed this software because we saw that it is needed by most of the people because when we write long essays then it is very difficult to find the mistakes. So in order to save the time we created this software. With the help of this software students will save there time. Mostly people make a lot of mistakes while writing essay or resumes. Our software will benefit students as well as a working professional. With the help of our software students can can easily find out the errors they generally commit while writing an essay or any paragraph. It will also help them to improve there writing capability. Freshers or working people who need to write resumes can use our software as it will help them to write there resumes without any error. Our main purpose is to help more and more people to write error free sentences or paragraph. Our software is user-friendly. So it can be used by any age group people. Software which are available in the market, charges a lot in order to correct the sentences. We are providing this for free to help students for their betterment.

Keywords:- *Spelling Corrector, Punctuation Corrector, Grammar Checker, Google Collab, Python.*

I. INTRODUCTION

The main objective is to automate identification of semantics error for enabling errorless proofreading. So with the help of Machine Learning algorithm we are detecting the error from a sentence or a paragraph. Mostly people make a lot of mistakes while writing essay or resumes. Mistakes which people commonly make while writing are spelling, misused, split and merge, subject verb agreement, common and proper, double words, comparative superlative, indefinite article, definite article, plurality, pronouns, consecutive nouns and tenses. We made a software which will help people to write sentences which are grammatically correct. Software that we have made will check the sentences in seconds. It will save the time as it takes a lot of time to recheck large essays or paragraph. We are providing this for free to help students for their betterment.[1]

We have named our software as “Grammar Checker”. When user enters the sentence then it highlight the error then we look for the correct words in the database. The database contains the correct words which we generally use in English language. The errors are checked in the database. The main function to test and corrects the grammar and then it returns the correct word for the error. At last the user gets the corrected text.

Grammar Checker can only correct the grammar of sentence or paragraph written in English language. The user should not input more than 600 characters at a time. The sentence should be written in utf-8 format. It cannot write digits instead of spelling out the numbers 0-9

II. RELATED WORK

A. *Methodology*

As illustrated in Figure 1, the types of error which can be in a sentence written in an English language are spelling, misused, split and merge, subject verb agreement, common and proper, double words, comparative superlative, indefinite article, definite article, plurality, pronouns, consecutive nouns and tenses.[1][2][3]

It corrects the word which has spelling errors, words which have a similar spelling, word which was accidentally split in two or vice-versa, two words were accidentally merged into a single word, the subject verb agreement in a given sentence, a proper or common noun which is not capitalized in a given sentence, repetition of a word in a given sentence, confusion between a/an when it is required in a given sentence, omission of the definite article ("the") in a given sentence, wrong usage of two or more nouns in a row in a given sentence in a given sentence, the confusion between the singular and plural form of a noun in a given sentence, the wrong pronoun given in a sentence, the wrong usage of comparative and superlative structures in a given sentence, the wrong verb tense in a given sentence, corrects the given sentence by applying the present simple tense, corrects the given sentence by applying the present perfect tense and corrects the given sentence by applying the present progressive tense.[6][7][8]

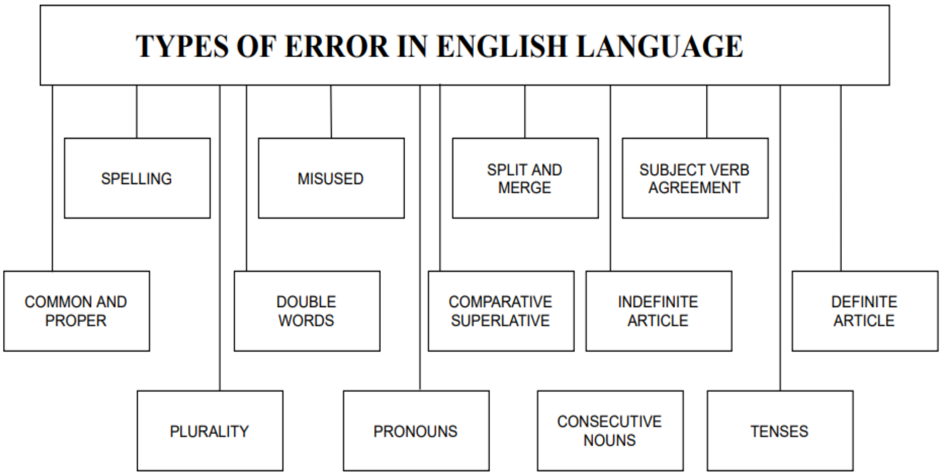

Figure 1 : Types of error in English language

B. *Solution Approach*

1) In this algorithm we are using database(ginger's api) to correct spelling, grammar, punctuation, vocabulary and style issues in documents.
2) The user should not input more than 600 characters at a time.
3) The sentence should be written in utf-8 format.
4) At the starting of the code we are importing all the necessary libraries which are required.
5) Then we create colorized output texts.
6) A function is created which uses ginger_api.
7) Then again we created a function which will get the sentences to the api.
8) Then at last we create a main function to test and correct our output.
9) Finally, check the text(original text) and then you will get the corrected text .
10) Errors in the original text will be highlighted in red color and the errors which are rectified in the corrected text will be highlighted in green color.This helps the user the understand the errors in the text. [1][4][5]

C. *Grammar Error types and metrics used for evaluation*

1) Spelling : It corrects the word which has spelling errors.
Original Text: The wooden statue had a big hed.
Corrected Text: The wooden statue had a big head.

2) Misused: It corrects the words which have a similar spelling.
Original Text: I want to rid a horse.
Corrected Text: I want to ride a horse.

3) Split and Merge: It corrects the word which was accidentally split in two or vice-versa, two words were accidentally merged into a single word.
Original Text: The bed room is beautiful.
Corrected Text: The bedroom is beautiful.

4) Subject Verb Agreement :It corrects the subject verb agreement in a given sentence .
Original Text: The smell of flowers bring back good memories.
Corrected Text: The smell of flowers brings back good memories.

5) Common and Proper Nouns : It corrects a proper or common noun which is not capitalized in a given sentence.
Original Text: monty is not well today.
Corrected Text: Monty is not well today.

6) Double Words : It corrects repetition of a word in a given sentence.
Original Text: I went to to the mall with Rishi.
Corrected Text: I went to the mall with Rishi.

7) Indefinite Article: It corrects confusion between a/an when it is required in a given sentence.

Original Text: Manish is studying for a MBBS degree.
Corrected Text: Manish is studying for an MBBS degree.

8) Definite Article: It corrects omission of the definite article ("the") in a given sentence .

Original Text: I had time of my life on this vacation.
Corrected Text: I had the time of my life on this vacation.

9) Consecutive Nouns: It corrects wrong usage of two or more nouns in a row in a given sentence in a given sentence.

Original Text:Riya went to buy ticket for us.
Corrected Text: Riya went to buy tickets for us.

10) Plurality: It corrects the confusion between the singular and plural form of a noun in a given sentence.

Original Text: We bought a number of item.
Corrected Text:We bought a number of items.

11) Pronouns: It corrects the wrong pronoun given in a sentence .

Original Text: I need you help.
Corrected Text: I need your help.

12) Comparative Superlative: It corrects the wrong usage of comparative and superlative structures in a given sentence.

Original Text: This movie is bad than anything I have seen.
Corrected Text:This movie is worse than anything I have seen.

13) Tenses: It corrects the wrong verb tense in a given sentence.

Original Text: It is important to submitted the paper on time.
Corrected Text: It is important to submit the paper on time.

14) Present Simple:It corrects the given sentence by applying the present simple tense.

Original Text: The battery is lasting for only three hours.
Corrected Text: The battery lasts for only three hours.

15) Present Perfect: It corrects the given sentence by applying the present perfect tense.

Original Text: I have develop good communication skills
Corrected Text:I have developed good communication skills

16) Present Progressive: It corrects the given sentence by applying the present progressive tense.

Original Text: Tom has writing a letter at the moment.
Corrected Text: Tom is writing a letter at the moment.

D. Software Requirements

1) In order to run our software we need a computer or laptop.
2) Our software will run on Google Collab.
3) A good internet connectivity is required to run our software.
4) Tools which are used to build are software is Python, ColoredText,Ginger Api.

E. Correcting Grammatical Errors with Machine Learning

To correct the grammatical errors of a given sentence.First we highlight the error then we look for the correct words in the database.The database contains the correct words which we generally use in English language.The errors are checked in the database and then it returns the correct word for the error. [1][13][14]

III. PROPOSED METHODOLOGY

F. Project Synopsis

1) We will import all the important libraries which are required to run the software.
2) Class ColoredText, here we colorize output text.
- Here we define a list with different colors.
- Create an empty dictionary.
- Using for loop we fill our dictionary(i.e. key value pairs.)
- def colorize:(cls(colour), text(word),color=None, bgcolor(background color)=None)
- C = Background color.
- try and except block is there for decoding with exceptional cases(i.e. words with color that we defined in our list)
- s_open, s_close=",," -- It is for opening and closing of a file which contains the paragraph or sentence.

3) get_ginger_url(Ginger api) --
- It is function which uses Ginger api. Here, Ginger api act as a database which contains words suggestions(in its correct spelling and grammatical form)
- Here, it uses the API_KEY, scheme, path, params, query and fragment.

4) get_ginger_result(text) --
- This function will get the sentences to the api.
- It means that with the help of this function we send our sentence to the ginger_api_url where it is checked for its grammatical and overall correctness.
- Again here, we use try and except blocks for dealing with exceptional cases(i.e. the sentences or word which isn't there in our database.)
- It checks the sentence and then prompt the message accordingly.(i.e. it can even raise an error message)

5) In the main function the message is received form get_ginger_result.
- If the grammar is correct then the color is changed to green and then it is printed on the terminal window.
- If error is received then the part of sentence or word is searched and then it send back to the previous functions for re-checking and correction.
- Finally, the corrected sentence is recieved and then gets printed as the output.

G. ALGORITHM

Step 1: START
Step 2: Import the libraries sys, urllib.parse, urllib.request, json, HTTPError and URLError.
Step 3: Create a class to colorize output texts.
Step 4: Create a function using Ginger API.
Step 5: Then again create a function to get the sentences to the Ginger API.
Step 6: Create a main function to test and correct our grammar.
Step 7: Enter the text and check the grammar of the given text.
Step 8: STOP

H. Design

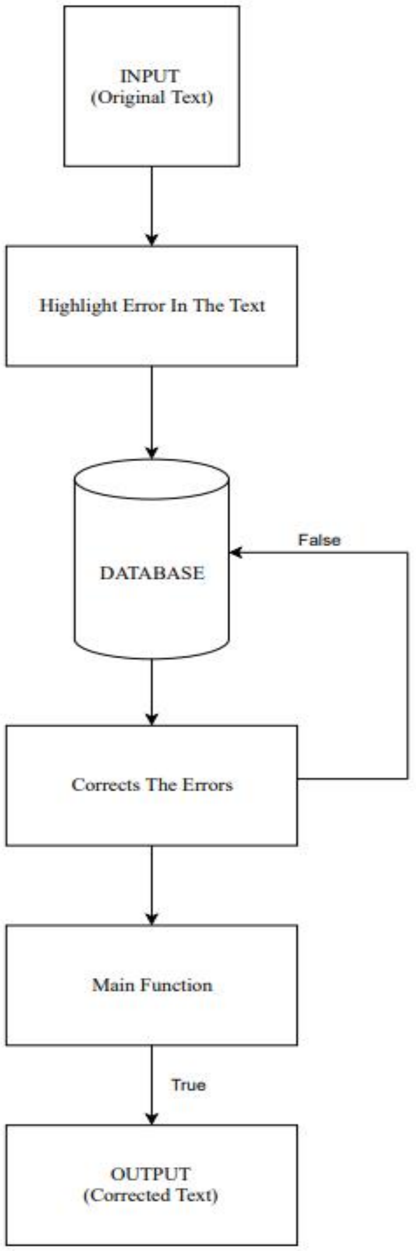

Figure 2 : Use Case Diagram for Grammar Checker

I. Implementation

We have implemented our software on Google Colab. We have used Python language as our project is based on Machine Learning.In the starting we have written the title of the project, our group number, name of the software and team members. Also we have wrote the main objective of project .[15][16]

As illustrated in Figure 2, we are importing the required libraries import sys, urllib.parse, urllib.request, json from urllib.error import HTTPError, from urllib.error import URLError

We are using Class ColouredText, here we colorize output text.

a. Here we define a list with different colors.
b. Create an empty dictionary.
c. Using for loop we fill our dictionary(i.e. key value pairs.)
d. def colorize: (cls(color), text(word), color=None, bgcolor(background color)=None)
e. C = Background color.
f. try and except block is there for decoding with exceptional cases(i.e. words with color that we defined in our list)
g. s_open, s_close=”,” -- It is for opening and closing of a file which contains the paragraph or sentence.[11][12]

Then we are using get_ginger_result(text) -- This function will get the sentences to the api.

a) It means that with the help of this function we send our sentence to the ginger_api_url where it is checked for its grammatical and overall correctness.
b) Again here, we use try and except blocks for dealing with exceptional cases(i.e. the sentences or word which isn’t there in our database.)
c) It checks the sentence and then prompt the message accordingly.(i.e. it can even raise an error message)[17][18]

Then we are using get_ginger_url(Ginger api)

a) It is function which uses Ginger api.Here, Ginger api act as a database which contains words suggestions(in its correct spelling and grammatical form)
b) Here, it uses the API_KEY, scheme, path, params, query and fragment.

At last we created the main function

a) In the main function the message is received form get_ginger_result.
b) If the grammar is correct then the color is changed to green and then it is printed on the terminal window.
c) If error is received then the part of sentence or word is searched and then it send back to the previous functions for re-checking and correction.
d) Finally, the corrected sentence is recieved and then gets printed as the output.

Then there is an option for the user to select color so we have selected red and green color.

a) The purpose of the red color is to highlight the error in the sentence which is the original text.
b) The green color highlight the corrected text.
c) The Original Text is the sentence or paragraph given by the user.
d) The Corrected Text is the sentence or paragraph with no error corrected with the help of Grammar Corrector.

Now we are typing the sentence or paragraph in the main to check whether there is any error.The Original Text has the sentence given by the user and Corrected Text has the sentence which is corrected. So the first sentence that we have checked is “The bed room is comfortable”

1) The mistake in the sentence is “bed room”. Bedroom is a single word so our Grammar Checker corrects the sentence.
2) The second sentence that we have checked is “The colour purple is my favorite”The mistake in the sentence is “colour”. The spelling of the word colour should be color.
3) The third sentence that we have checked is “My friend john is not well”The mistake in the sentence is “john”. The word is a name of a person so the first letter should be in capital letter.
4) The fourth sentence that we have checked is “I went to to the store”The mistake in the sentence is “to to”.Grammar checker corrects the repeated word .[19][20]

J. Limitations

1) It can only correct the grammar of sentence or paragraph written in English language.
2) The user should not input more than 600 characters at a time.
3) The sentence should be written in utf-8 format.
4) It cannot write digits instead of spelling out the numbers 0-9 .

K. Future Scope of the Project

1) We cannot check more than 600 characters at a time .So we need to exceed the character limit.
2) It cannot completely correct the grammar of the sentence. Therefore we need enhance it.
3) It can only correct the grammar of sentence or paragraph written in English language. So we need to correct the grammar for other languages.
4) We need to add a feature named “Numeral Spelling Out” which write digits instead of spelling out the numbers 0-9.
5) The sentence should be written in utf-8 format so we need to add other writing format.[9][10]

IV. CONCLUSION

It first takes input from the user, which is the “Original Text” and the detects the grammatical errors and contextual errors from it and then gives the “Corrected Text”.The “Original Text” can be a sentence or a paragraph.

Example:

1) Original Text : english is not so that very eassy.
Corrected Text: English is not so that very easy.
2) Original Text : I am nnnot okka.
Corrected Text: I am not okay.
3) Original Text : wer do you go to scul?
Corrected Text: Where do you go to school?
4)Original Text : good for people who are English learners.
Corrected Text: Good for people who are English learners.

REFERENCES

[1] AS Hornby. 1995. Guide to Patterns and Usage in English. (1995)

[2] John Lee and Stephanie Seneff. 2008. Correcting Misuse of Verb Forms.. In ACL. 174–182

[3] Daniel Naber. 2003. A rule-based style and grammar checker. (2003).

[4] Alla Rozovskaya, Kai-Wei Chang, Mark Sammons, and Dan Roth. 2013. The University of Illinois system in the CoNLL-2013 shared task. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task. 13–19.

[5] Joachim Wagner, Jennifer Foster, and Josef van Genabith. 2007. A comparative evaluation of deep and shallow approaches to the automatic detection of common grammatical errors. Association for Computational Linguistics

[6] PC Wren and H Martin. 2000. English Grammar & Composition. S. Chand & Company Ltd (2000).

[7] Chak Yan Yeung and John Lee. 2015. Automatic Detection of Sentence Fragments.. In ACL (2). 599–603.

[8] Zheng Yuan. 2017. Grammatical error correction in non-native English. Technical Report. University of Cambridge, Computer Laboratory

[9] A. M. Wing, and A. D. Baddeley, Spelling errors in handwriting: a corpus and distributional analysis, Cognitive processes in spelling, London: Academic Press, pp.251-285, 1980

[10] Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, Jingming Liu,Improving Grammatical Error Correction via Pre-Training a Copy-Augmented Architecture with Unlabeled Data,Yuanfudao Research / Beijing, China ,2019

[11] Emily M Bender, Dan Flickinger, Stephan Oepen, Annemarie Walsh, and Timothy Baldwin. 2004. Arboretum: Using a precision grammar for grammar checking in CALL. In Instil/icall symposium 2004

[12] Chris Brockett, William B Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal SMT techniques. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 247–256.

[13] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning, pages 160–167. ACM.

[14] Daniel Dahlmeier and Hwee Tou Ng. 2011. Grammatical error correction with alternating structure optimization. In Proceedings of the 49th Annual Meeting

of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 915–923

[15] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In Advances in Neural Information Processing Systems, pages 6294–6305.

[16] Bestgen, Yves, and Sylviane Granger. "Categorising spelling errors to assess L2 writing." International Journal of Continuing Engineering Education and Life-Long Learning 21.2-3 (2011): 235- 252

[17] Peters, Matthew E., et al. "Deep contextualized word representations." arXiv preprint arXiv:1802.05365 (2018).

[18] Staffs Keele et al. 2007. Guidelines for performing systematic literature reviews in software engineering. In Technical report, Ver. 2.3 EBSE Technical Report. EBSE. sn.

[19] Akshat Kumar and Shivashankar Nair. 2007. An artificial immune system based approach for English grammar checking. Artificial immune systems (2007), 348–357

[20] Guihua Sun, Xiaohua Liu, Gao Cong, Ming Zhou, Zhongyang Xiong, John Lee, and Chin-Yew Lin. 2007. Detecting erroneous sentences using automatically mined sequential patterns. In ACL. 23–30.