GRAMMAR CHECKER

PSAT MINI-PROJECT PRESENTATION

Presented by C S AMRITHA
CB.EN.U4CYS22016
TIFAC-CORE in Cyber Security
Amrita Vishwa Vidyapeetham, Coimbatore Campus

Feb 24, 2023

AMRITA VISHWA VIDYAPEETHAM

Table of Contents

Introduction

Grammar Checker

A grammar checker will find each sentence in a text, look up each word in the dictionary, and then attempt to change the sentence into a form that matches a grammar.

- The program may then identify numerous mistakes, such as agreement in tense, number, word order, and so on, using a variety of rules. Additionally, several stylistic issues with the text are also detectable.
- Grammar checkers may attempt to identify passive sentences and suggest an active-voice alternative.
- Examples :

  Google Docs and Sapling.ai (built into systems)
  Grammarly and Qordoba (broweser extensions)
  Ginger(desktop application)
  LanguageTool(open source software)

Figure: Grammar Checker

Problem Analysis And Dissection I

Coding a grammar checker can be fairly challenging as as it involves analyzing text at both the syntactic and semantic level.During the development process, it is important to address a number of important issues, including:

1. **Tokenization**: The text needs to be broken down into individual words and parts of speech so that grammar rules can be applied.

2. **Grammar rules**: A set of grammar rules needs to be defined and implemented in the program. These rules need to cover a wide range of grammatical errors, such as subject-verb agreement, verb tense, and punctuation.

3. **Contextual analysis**: To recognise and fix grammatical faults that might be confusing or context-dependent, the programme needs to be able to comprehend the text's context.

4. **Error detection**: The program needs to be able to accurately identify grammatical errors and flag them for correction.

Problem Analysis And Dissection II

5. **Error correction**: The program should be able to suggest corrections for identified errors, or automatically correct them.

6. **Scalability**: The program should be able to handle large volumes of text, and be able to process text efficiently.

7. **Human-like understanding**: The program should be able to understand idiomatic expressions, colloquialism and sarcasm, which human-written text often contain.

Basic Rules For A Sentence To Be Syntactically Correct

1. Sentence must begin with a uppercase character.
2. Then lowercase characters follow.
3. There must be spaces between each words.
4. Sentence must end with a full stop.
5. Two continuous spaces are invalid.
6. Two continuous uppercase characters are not allowed.

Algorithm

The steps for a algorithm for grammar checking are as follows:

1. Tokenize the input text into words and punctuation marks.
2. Perform part-of-speech tagging to assign a grammatical category to each word.
3. Check the sentence for errors by applying a set of predefined grammar rules. These rules may include checks for subject-verb agreement, verb tense, word order, and punctuation.
4. Identify any errors that are found and provide suggestions for corrections.
5. Repeat the process for each sentence in the input text.

General Limitations of Grammar Checker Program

Grammar Check is only a program,it may miss most of the things expecially while dealing with idioms.It only checks sentences and will not produce better ideas nor group your ideas into a logical sentence.

- **Context Based**: Your sentence won't be changed word for word. Results are totally dependant on context.
- **Wrong application**: Complete failure when it comes to incorrect usage of words.It may be grammatically correct but the sentence as a whole may not make sense.
- **Checking Proper Nouns**: Detecting proper nouns using most grammar checkers is not feasible.On finding proper nouns,initial letter of the word should automatically be capitalized.

Limitations of Grammar Checker in Flogorithm

In general, flowgramming is better suited for straightforward procedural tasks like data manipulation and basic calculations.

For a grammar check program, which requires natural language processing and understanding of context, a flowgramming approach may not be the best option.

Flowgorithm do not have the capability to understand natural language and process it, which is required for grammar checking.

Additionally, A flowgramming approach would not be able to handle some of the complex and nuanced tasks that a grammar check program requires such as understanding context, idiomatic expressions and sarcasm.

Thank you

Thank You!

References

https://openai.com/blog/chatgpt
https://digital.com/best-grammar-checker/
https://www.wikipedia.org/
https://blog.hubspot.com/marketing/best-grammar-checker