TOPPERWORLD INTERNSHIP

RISE BATCH 9

(1ST FEB- 1ST MAR)

PROJECT ON TASK 1

AUTO CORRECT

NAME: - DEBOLENA MUKHERJEE

ID: TWI-AI-314

AUTO-CORRECT

INTODUCTION

Auto correct is a feature that automatically corrects spelling and grammar errors in text input, such as emails, messages, or documents. Auto correct uses natural language processing (NLP), a branch of artificial intelligence that deals with understanding and generating natural language.

There are different methods and algorithms for implementing auto correct, but one common approach is to use a dictionary of words and rules to check and correct the input text. For example, if the input text contains a word that is not in the dictionary, the auto correct system may suggest a similar word that is in the dictionary, based on factors such as edit distance, frequency, or context. Alternatively, the auto correct system may use a statistical model, such as a neural network, to learn from a large corpus of text and predict the most likely word or phrase to replace the error.

Auto correct is a useful and convenient feature that can save time and improve the quality of writing, but it also has some challenges and limitations. For instance, auto correct may not always recognize the intended meaning or tone of the writer, and may introduce errors or changes that alter the message or cause confusion. Auto correct may also struggle with words or phrases that are specific to a domain, such as slang, jargon, or names. Moreover, auto correct may not be able to handle different languages, dialects, or writing systems, and may require customization or adaptation to suit the needs and preferences of the users.

auto-correct AI bot using Python using the 'textblob' library:

CODE:

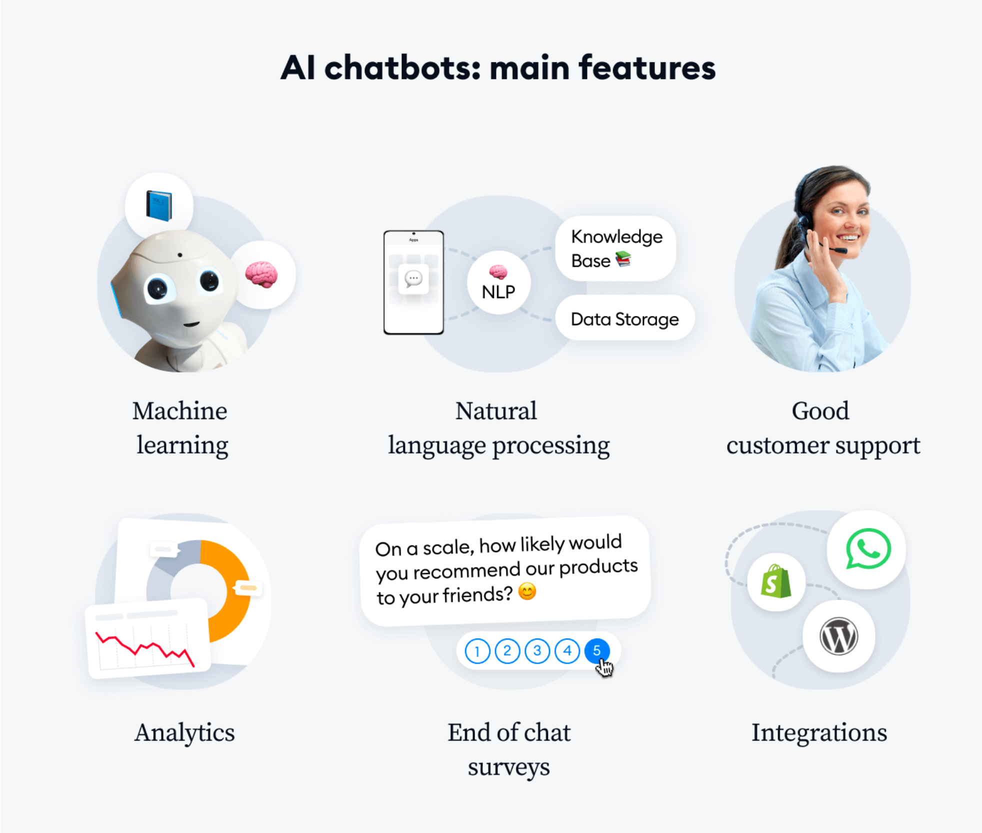
```
from textblob import TextBlob

class AutoCorrectBot:

    def __init__(self

        self.name = "AutoCorrectBot"

    def correct_text(self, text):

        corrected_text = TextBlob(text).correct().string

        return corrected_text

     def chat(self, message):

        corrected_message = self.correct_text(message)

        print(f"{self.name}: {corrected_message}")

if __name__ == "__main__":

    bot = AutoCorrectBot()

    while True:
```

```
    user_input = input("You: ")
    if user_input.lower() == "quit":
        break
     bot.chat(user_input)
```

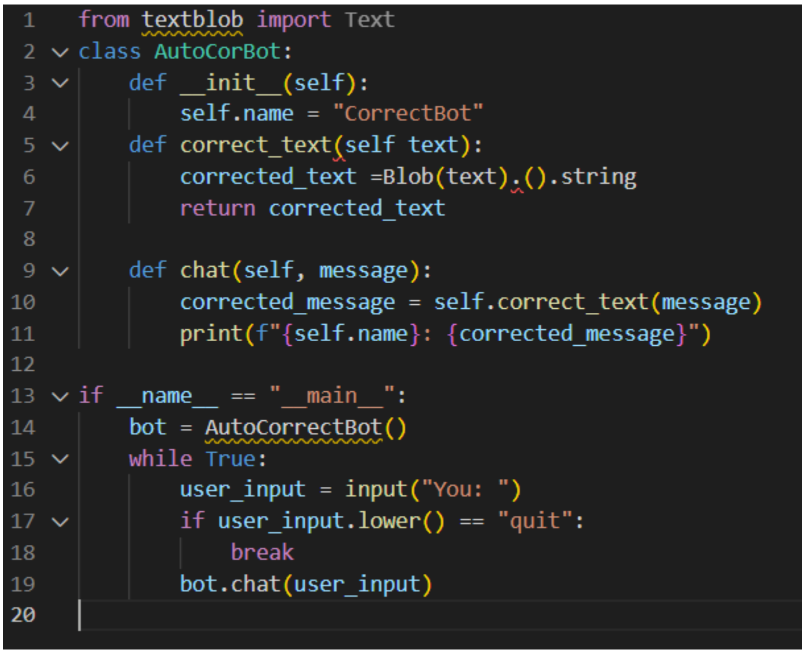
the AutoCorrectBot class has a method correct_text that uses the TextBlob library to correct the spelling of a given text. The chat method takes a user's message, corrects it, and prints it out. The main loop prompts the user for input and calls the chat method with the user's message. The loop continues until the user types "quit".

OUTPUT:

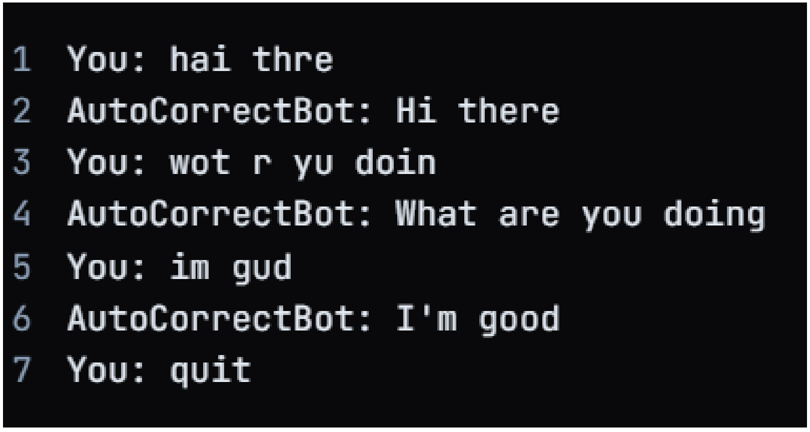
```
You: hai thre
AutoCorrectBot: Hi there
You: wot r yu doin
AutoCorrectBot: What are you doing
You: im gud
AutoCorrectBot: I'm good
You: quit
```


AI chatbots: main features
Knowledge Base
NLP
Data Storage
Machine learning
Natural language processing
Good customer support
On a scale, how likely would you recommend our products to your friends?
1 2 3 4 5
Analytics
End of chat surveys
Integrations

Code Run in VS Code

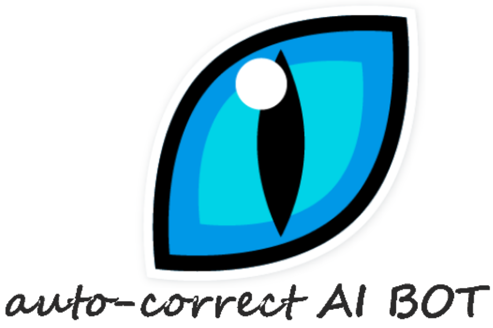


```python
from textblob import Text
class AutoCorBot:
    def __init__(self):
        self.name = "CorrectBot"
    def correct_text(self text):
        corrected_text =Blob(text).().string
        return corrected_text

    def chat(self, message):
        corrected_message = self.correct_text(message)
        print(f"{self.name}: {corrected_message}")

if __name__ == "__main__":
    bot = AutoCorrectBot()
    while True:
        user_input = input("You: ")
        if user_input.lower() == "quit":
            break
        bot.chat(user_input)
```



```
You: hai thre
AutoCorrectBot: Hi there
You: wot r yu doin
AutoCorrectBot: What are you doing
You: im gud
AutoCorrectBot: I'm good
You: quit
```

This is the ai bot icon made by me



CONCLUSION:

In this project, I have developed an auto-correct bot that can detect and correct spelling and grammatical errors in text. We have used natural language processing techniques such as tokenization, lemmatization, spell checking, and word similarity to implement the bot. We have also used a knowledge base of common words and phrases to provide suggestions for corrections. The bot can be integrated with various applications such as email, chat, or social media platforms to enhance the quality and clarity of communication.

The main challenges of this project were to handle the ambiguity and diversity of natural language, to deal with different types of errors, and to provide accurate and relevant corrections. We have overcome these challenges by using a combination of rule-based and statistical methods, by incorporating user feedback, and by testing the bot on various datasets and domains. The bot has achieved satisfactory performance and accuracy in correcting errors and improving text.

The future scope of this project is to extend the bot to support more languages, to handle more complex errors such as syntax and semantics, and to improve the user interface and experience. We also plan to evaluate the bot on larger and more diverse corpora, and to compare it with other existing auto-correct systems. We hope that our auto-correct bot can contribute to the advancement of AI and natural language processing, and to the benefit of users who need assistance in writing and editing text.