Minor Project Report

Submitted in partial fulfillment of the degree of

B. Tech

Computer Science & Engineering

By

Aditya Singh(11900121095),
Karmanya Thapa(11900121096),
Dayanand Jajodia(11900121094),
Akansha Rani(11900121093),
Ritu Sarkar(11900121097).

Third year student of

Siliguri Institute of Technology

Under the supervision of

Mr. Kumarjeet Gupta
Sikharthy Infotech Pvt. Ltd.

Department of Computer Science & Engineering

Date:

I hereby forward the documentation prepared by me **Aditya Singh, Karmanya Thapa, Dayanand Jajodia, Akansha Rani, Ritu Sarkar** under the supervision of Mr. Kumarjeet Sir entitled **Auto-correct Feature Using NLP** accepted as fulfilment of the requirement for the Degree of Bachelor of Technology in **Computer Science & Technology** (B.Tech) from **Siliguri Institute of Technology** affiliated to **Maulana Abul Kalam Azad University of Technology** (**MAKAUT**).

_____

Mr. Kumarjeet Gupta
(Senior Software Engineer &
Project Manager)

Project Guide

Sikharthy Infotech Pvt. Ltd.

Auto-Correct Feature Using NLP

By

Aditya Singh(11900121095),
Karmanya Thapa(11900121096),
Dayanand Jajodia(11900121094),
Akansha Rani(11900121093),
Ritu Sarkar(11900121097).

UNDER THE GUIDANCE OF

Mr. Kumarjeet Gupta

Project Guide

Sikharthy Infotech Pvt. Ltd.

THIS IS SUBMITTED IN FULFILLMENT OFTHE REQUIREMENTS FOR THE DEGREE OF

B.TECH

IN

COMPUTER SCIENCE & ENGINEERING

SILIGURI INSTITUTE OF TECHNOLOGY

AFFILIATED TO
Maulana Abul Kalam Azad University of Technology

Address: Hill Cart Road, Salbari, Sukna, West Bengal 734009

Phone: 094345 27272

Website: https://sittechno.org/

Certificate of Approval

The foregoing project is hereby approved as a creditable study for the B. Tech in Computer Science & Engineering and presented in a manner of satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approved any statement made, opinion express or conclusion therein but approve this project only for the purpose for which it is submitted.

Final Examination for
Evaluation of the Project

Signatures of Examiners

ABSTRACT

The purpose of the project entitled as “Auto-Correct Feature Using NLP” is to develop a system which is user friendly simple, fast, and cost effective. It deals with the detection of typing errors made by users and suggests corrections to them. The main function of the system is to correct typos (Typing Errors). This makes use of various features of NLP (Natural Language Processing) and a data-set and finds out the most suitable suggestions to correct a typing error.

ACKNOWLEDGEMENT

It is a great pleasure for me to acknowledge the assistance and participation of a large number of individuals to this attempt. Our project report has been structured under the valued suggestion, support and guidance of **Mr. Kumarjeet Gupta**. Under his guidance we have accomplished the challenging task in a very short time.

Finally, we express our sincere thankfulness to our family members for inspiring me all throughout and always encouraging us.

TABLE OF CONTENTS

INTRODUCTION

In the realm of digital communication, typos and spelling mistakes are ubiquitous nuisances. Auto-correct, a feature widely integrated into modern devices and applications, seeks to alleviate this issue by automatically suggesting or correcting misspelled words as users type. Leveraging the power of Natural Language Processing (NLP), auto-correct algorithms analyze and interpret text input, aiming to predict the intended word based on context and linguistic patterns.

1.1 Python

We used Python in this project we learned python datatypes, methods, class, sorting, OOPs concept, loops and many elements in our python program.

1.2 IDE

We used Microsoft Visual Studio Code (VS Code) in our project as IDE.

PURPOSE

The primary objective of auto-correct is to enhance user experience by minimizing typing errors and facilitating faster, more accurate communication. By swiftly correcting mistakes as users type, auto-correct streamlines the process of composing messages, emails, documents, and other forms of text input. Moreover, it serves as a valuable tool for individuals with varying levels of spelling proficiency, ensuring that their messages convey the intended meaning effectively.

FUNCTIONALITY

Auto-correct operates through a sophisticated interplay of algorithms and linguistic databases. As users input text, the auto-correct system continuously evaluates the sequence of characters, comparing them against a comprehensive dictionary of words and their respective frequencies. Additionally, it considers contextual cues, such as adjacent words, language patterns, and common phrases, to generate accurate correction suggestions.

4.1 IDENTIFY THE MISSPELLED WORD

A word is identified as a typo if it is not present in the software's vocabulary. The vocabulary is in the form of a set with unique elements.

4.2 FIND STRINGS THAT ARE N EDIT CHANGES AWAY

n stands for the number of characters to edit to transform the observed string into a target string. The observed string, in our case, is the typo, while the target string is the potential candidate for the suggestion.

There are four types of manipulations that change our string into a potential sensible suggestion:

Insert (add a letter): “helo” -> “hello”
Remove (remove a letter): “hellol” -> “hello”
Exchange (Swap one letter with another): “hlelo” -> “hello”
Replacement (Add a letter at position k, removing the original letter at position k): “hillo” -> “hello”
Each of these manipulations transforms an incorrect string into a meaningful string. Obviously the computer has no idea which word is full or not, so m elements are generated for each of the manipulations in the example. This creates a very long list of potential candidates which must then be filtered to remove gibberish.

4.3 FILTER CANDIDATES

Now, we have to find the intersection between the set of vocabulary and the set of manipulated strings.

4.4 COMPUTE THE PROBABILITIES OF SUGGESTIONS

The last step is to select the candidate who has the highest probability of occurring in that given context. This is how a “base” model of autocorrect can provide a valid hint for our error. The mathematical formula for calculating the probabilities that a word may occur in our corpus is as follows

$$P(word) = \frac{C(word)}{V}$$

the probability *P* that a word appears in the corpus is equal to how many times we see that word in our vocabulary over the length of our vocabulary *V*. Let's take the following example:

Word	Count
I	10
am	7
good	1
with	4
Python	2
and	6
R	1

Assuming that our corpus is made up of 165 terms, we see how the word *Python* has a probability of occurring of 2 / 165 = 0.012.

We now have a probability distribution of the words in our corpus. Our autocorrect system will then select the candidate who has the highest probability of occurring in our corpus. Very simple.

REQUIREMENTS

5.1 DATASET

A dataset, consisting of a set of vocabulary words against which the input word is checked against, is obtained from Kaggle.

5.2 REGULAR EXPRESSION (re) MODULE

A Regular Expression or RegEx is a special sequence of characters that uses a search pattern to find a string or set of strings. It can detect the presence or

absence of a text by matching it with a particular pattern and also can split a pattern into one or more sub-patterns.
Python has a built-in module named “re” that is used for regular expressions in Python. We can import this module by using the import statement.

```
import re
```

5.3 COUNTER FROM PYTHON COLLECTIONS

The collection Module in Python provides different types of containers. A Container is an object that is used to store different objects and provide a way to access the contained objects and iterate over them. Some of the built-in containers are Tuple, List, Dictionary, etc. In this article, we will discuss the different containers provided by the collections module.
A counter is a sub-class of the dictionary. It is used to keep the count of the elements in an iterable in the form of an unordered dictionary where the key represents the element in the iterable and value represents the count of that element in the iterable.

```
from collections import Counter
```

5.4 HARDWARE REQUIREMENTS

The minimum Hardware requirements for the application to run smoothly should have the following configuration:

Processor	Intel Core i3
RAM	4GB or more
HDD	3GB or more

NLP TECHNIQUES USED

Natural language processing (NLP) is a subfield of artificial intelligence (AI) that deals with the interaction between humans and computers using natural language. NLP is concerned with developing algorithms and computational models that enable computers to understand, analyze, and generate human language.

6.1 TOKENIZATION

In natural language processing (NLP), tokenization is the process of breaking up a stream of text, typically a sentence or document, into smaller units called "tokens". These tokens are typically words, but can be sub-words or characters depending on the granularity required for a particular NLP task. Tokenization is a fundamental step in NLP as it lays the foundation for further analysis and processing of text data. The main goal of tokenization is to break up a continuous stream of text into discrete units that computers can easily process and analyze. By segmenting the text into tokens, NLP models and algorithms can understand the individual parts of the text and extract meaningful information from them.

6.2 SPELL CHECKING

This is a technique used in NLP (Natural Language Processing). Here, we identify the misspelled words against a corpus of correctly spelled words (Dataset).

6.3 CALCULATING PROBABILITY

We already have a dataset of correctly spelled words. We find the probability of occurrence of each of these words. While intersection, out of a list of intersected data, the three words with the best probability is chosen.

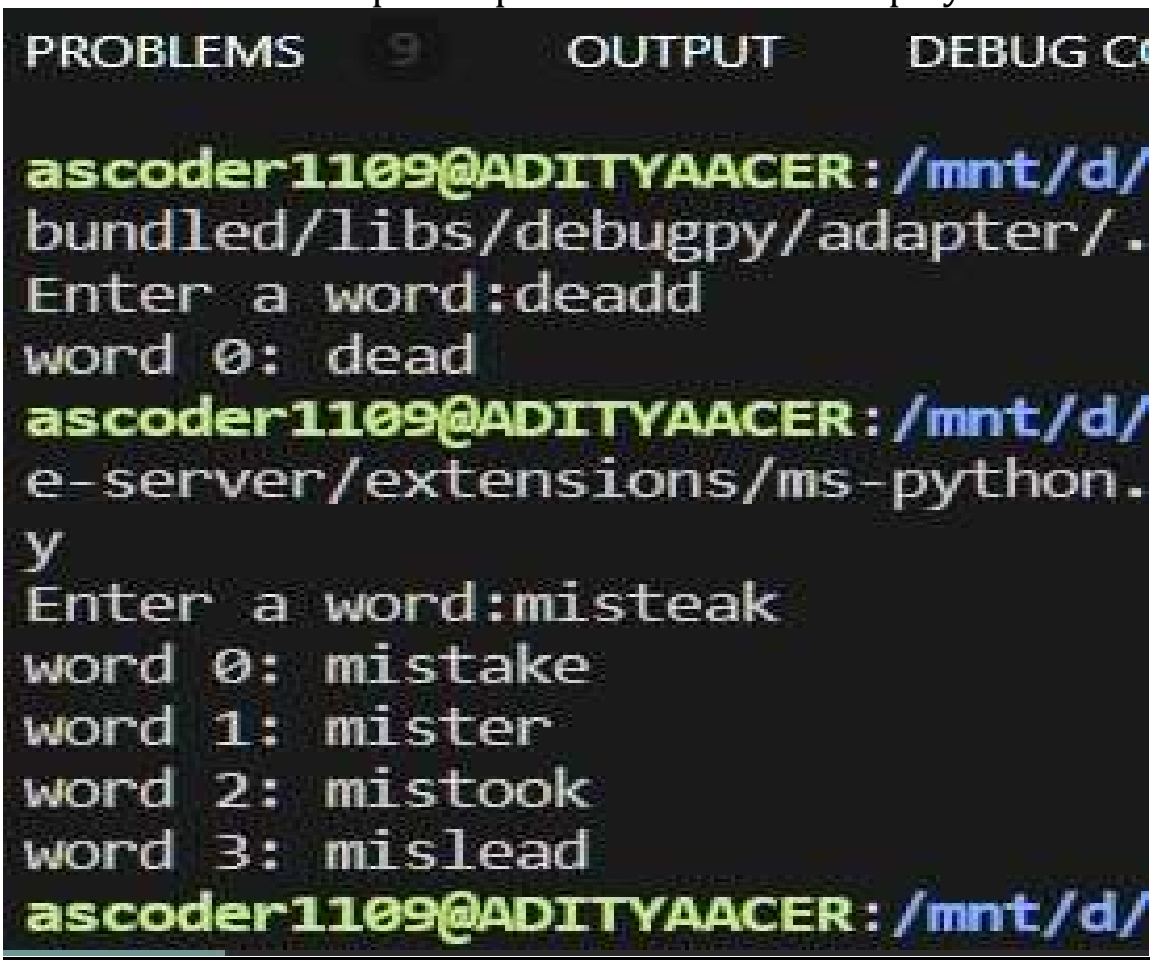
6.4 INTERSECTION

The word that the user gives as input is checked against the set of vocabulary we fed the model as our dataset using the intersection function. This list of resulting values is saved in a list. Out of that list, three words with the best probability is selected and displayed.

RESULT ANALYSIS

The main objective of this project is to make a model which can detects mistakes in a word and ultimately gives us one or a list of correct suggestions corresponding to the input. The model that we created gives three word with the best probabilities of occurrence.

Given below is a sample output that our model displays:



LIMITATIONS

Although our model can successfully display the correct suggestions for a misspelled word, it has its own limitaions.

- This model is not able to perform auto-corrections in a sentence.
- This model cannot perform checks and corrections for grammatical errors.
- Time Complexity of this algorithm is too high as the list of words is traversed repeatedly.

CONCLUSION

The project “AUTOCORRECT FEATURE USING NLP” is for detecting spelling errors and suggesting corrections. Users can access their stud account and see/download results. Through continuous learning from vast amounts of linguistic data, autocorrect algorithms become increasingly adept at understanding user intent and context, leading to more precise and contextually relevant corrections. This not only saves users time and effort but also helps prevent misunderstandings and miscommunication in digital conversations. Furthermore, the autocorrect feature demonstrates the practical applications of NLP in everyday technology, showcasing its potential to streamline and enhance human-computer interaction. As NLP continues to advance, we can expect autocorrect systems to become even more sophisticated, catering to diverse languages, dialects, and writing styles with higher accuracy and adaptability.

REFERENCES

1. Dataset Link: https://www.kaggle.com/datasets/bittlingmayer/spelling
2. .https://www.geeksforgeeks.org/autocorrector-feature-using-nlp-in-python/
3. https://pianalytix.com/autocorrect-using-nlp/