A Project/Dissertation Review Report

on

Automate Identification of semantic errors for enabling errorless proof reading

Submitted in partial fulfillment of the

requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Under The Supervision of

Name of Supervisor : Dr S Jerald Nirmal Kumar

Designation : Associate Professor

Submitted by

YASH GARG - 18SCSE1010083

VIBHU MISHRA - 18SCSE1010052

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GREATER NOIDA, UTTAR PRADESH
Winter 2021 – 2022

SCHOOL OF COMPUTING SCIENCE ANDENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **" Automate Identification of emantic Errors for enabling errorless proof reading"** in partial fulfillment of the requirements for the award of the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** submitted in the **School of Computing Science and Engineering** of Galgotias University, Greater Noida, is an original work carried out during the period of **JAN 2022 to MAY 2022**, under the supervision of **Mr. Dr S Jerald Nirmal Kumar, Associate Professor**, **Department of Computer Science and Engineering** of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any other places.

18SCSE1010052 – VIBHU MISHRA

18SCSE1010083 – YASH GARG

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor

(Dr S Jerald Nirmal Kumar)

Associate Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **18SCSE1010052 – VIBHU MISHRA, 18SCSE1010083 – YASH GARG** has been held on…….. and his/her work is recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**.

Signature of Examiner(s) Signature of Supervisor(s)

Signature of Project Coordinator **Signature of Dean**

Date:

Place:

Abstract

My objective is to develop **Machine Learning** algorithms with an aim to detect the grammaticalerrors and contextual errors from a given sentence or paragraph and then recognize the same.

- A Grammar Checker is software or a program which is used to find grammatical errors froma given sentence or a paragraph. That is to say, it checks for improper sentence structure and word usage (e.g., their instead of there), poorly placed or unnecessary punctuation, and other more esoteric errors.

- The Grammar Checker helps you write better English and efficiently corrects texts. Based on the context of complete sentences, Grammar Checker uses Machine Learning to correct grammar mistakes, spelling mistakes and misused words, with unmatched accuracy. Grammar check software improves your text just like a human reviewer would.

- Grammar checkers underlines the mistakes given in sentence or a paragraph and Ginger uses groundbreaking technology to detect grammar and spelling errors in sentences and to correct them with unmatched accuracy. From singular vs plural errors to the most sophisticatedsentence or tense usage errors, Ginger picks up on

mistakes and corrects them.

- Grammar Checker uses Ginger API tool to correct all types of mistakes which present in a sentence or a paragraph written in English language. Ginger corrects all types of grammatical mistakes including topics that are not addressed by any other grammar correction program.

Here are some examples:

Subject verb agreement

The smell of flowers **bring** back memories.

→ The smell of flowers **brings** back memories.

Singular/Plural nouns

Six people lost their **life** in the accident

→ Six people lost their **lives** in the accident.

Consecutive nouns

Sheryl went to the **tickets** office

→ Sheryl went to the **ticket** office.

Misused words correction

Using its contextual grammar checker, Ginger recognizes the misused words in any sentence andreplaces them with the correct ones.

I was **wandering** if there's any news.

→ I was **wondering** if there's any news.

Contextual spelling correction

The Ginger Spell Checker is a contextual spell checker which identifies the correction that bestfits the meaning of the original sentence. When combined with the Ginger Grammar Checker, you

can correct entire sentences in a single click. The same misused word will have a differentcorrection based on the context:

The marble statue **hed** a big **hed**

→ The marble statue **had** a big **head**.

Phonetic spelling mistakes are corrected even if the correct spelling is very different from the waythey were originally written:

I like books, **exspecaley** the classics

→ I like books, **especially** the classics

Irregular verb conjugations are corrected as well:

He **flyed** to Vancouver

→ He **flew** to Vancouver

CHAPTER 1

INTRODUCTION

1.1 Objective of the project

My objective is to develop **Machine Learning** algorithms with an aim to detect the grammatical errors and contextual errors from a given sentence or paragraph and then recognize the same. A Grammar Checker is software or a program which is used to find grammatical errors from a given sentence or a paragraph. That is to say, it checks for improper sentence structure and word usage (e.g., their instead of there), poorly placed or unnecessary punctuation, and other more esoteric errors.

1.2 Problem statement and research objectives

The Grammar Checker helps you write better English and efficiently corrects texts. Based on the context of complete sentences, Grammar Checker uses Machine Learning to correct grammar mistakes, spelling mistakes and misused words, with unmatched accuracy. Grammar check software improves your text just like a human reviewer would.Grammar checkers underlines the mistakes given in sentence or a paragraph and Ginger uses groundbreaking technology to detect grammar and spelling errors in sentences and to correct them with unmatched accuracy. From singular vs plural errors to the most sophisticated sentence or tense usage errors,

Ginger picks up on mistakes and corrects them.

Grammar Checker uses Ginger API tool to correct all types of mistakes which present in a sentence or a paragraph written in English language. Ginger corrects all types of grammatical mistakes including topics that are not addressed by any other grammar correction program.

1.3 Description of Domain

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications tobecome more accurate at predicting outcomes without being explicitly programmed to do so.

Machine learning algorithms use historical data as input to predict new output values.

The term *machine learning* was coined in 1959 by Arthur Samuel, an American IBMer and pioneer in the field of computer gaming and artificial intelligence. Also the synonym *self-teaching computers* was used in this time period. A representative book of the machine learning research during the 1960s was the Nilsson's book on Learning Machines, dealing mostly with machine learning for pattern classification. Interest related to pattern recognition continued into the 1970s, as described by Duda and Hart in 1973. In 1981 a report was given on using teaching strategies so that a neural network learns to recognize 40 characters (26 letters, 10 digits, and 4 special symbols) from a computer terminal.

Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: "A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P* if its

performance at tasks in *T*, as measured by *P*, improves with experience *E*." This definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing's proposal in his paper "Computing Machinery and Intelligence", in which the question "Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do?".

Modern day machine learning has two objectives, one is to classify data based on models which have been developed, the other purpose is to make predictions for future outcomes based on these models. A hypothetical algorithm specific to classifying data may use computer vision of moles coupled with supervised learning in order to train it to classify the cancerous moles. A machine learning algorithm for stock trading may inform the trader of future potential predictions.

CHAPTER 2

TECHNICAL DESCRIPTION

Solution Approach:

1. In this algorithm we are using Ginger's proofreading 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.

Project Synopsis:

1. In the starting of our code, we are importing the required libraries.
2. Class ColouredText, 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 wedefined in our list)

*s_open, s_close=”,” -- It is for opening and closing of a file which contains the paragraph orsentence.

3. Get_ginger_url(Ginger api) -- It is function which uses Ginger api.Here, Ginger api act as adatabase 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 ischecked for its grammatical and overall correctness.

*Again here, we use try and except blocks for dealing with exceptional cases(i.e. the sentences orword which isn't there in our database.)

*It checks the sentence and then prompt the message accordingly.(i.e. it can even raise an errormessage)

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 terminalwindow.

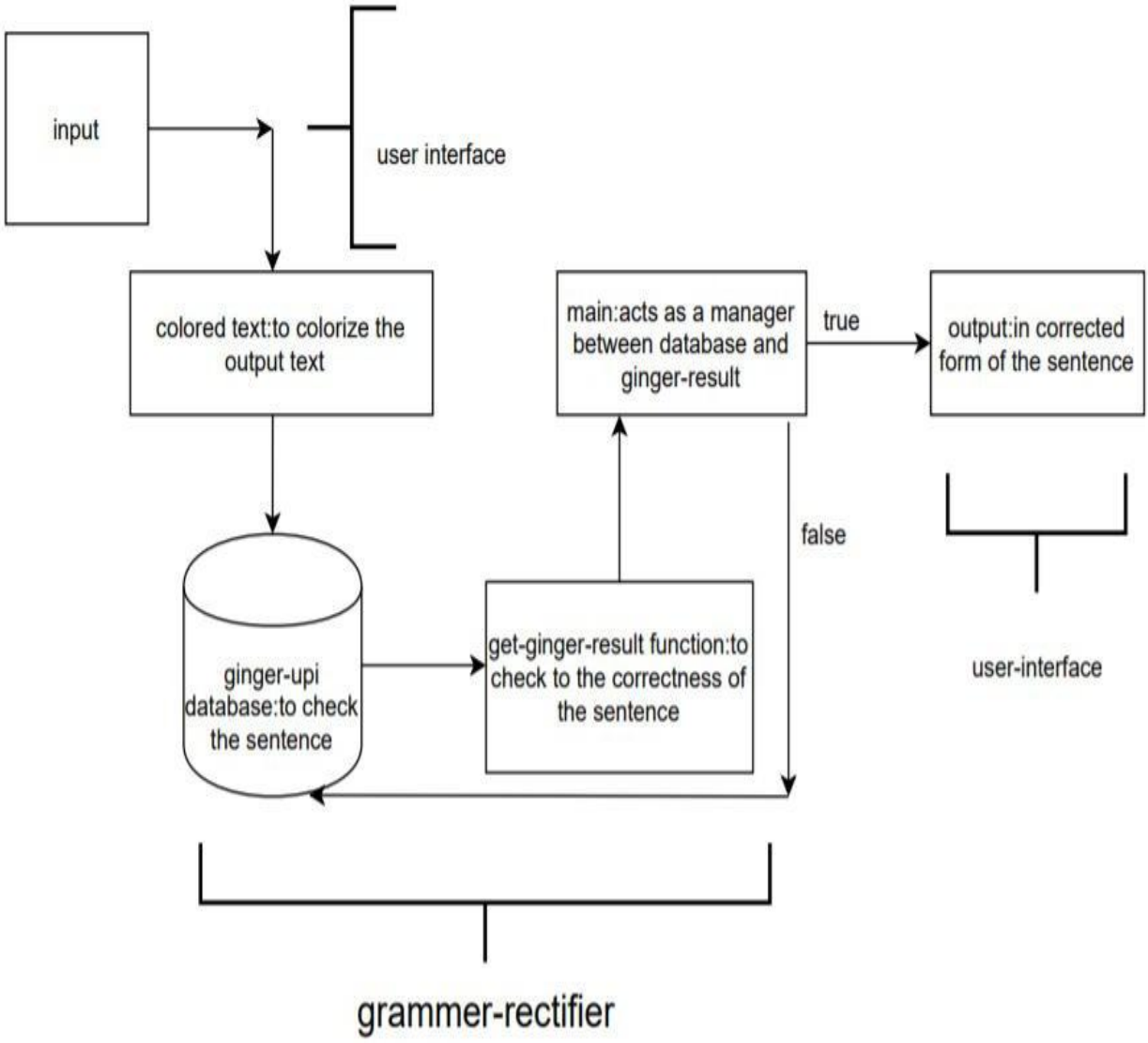
*If error is received then the part of sentence or word is searched and then it send back to theprevious functions for re-checking and correction.

*Finally, the corrected sentence is recieved and then gets printed as the output.

CHAPTER 3

SYSTEM

DESIGN



CHAPTER 4

SYSTEM IMPLEMENTATION

Solution Approach:

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

Project Synopsis:

1. In the starting of our code, we are importing the required libraries.

2.Class ColouredText, 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 wedefined in our list)

*s_open, s_close=”,” -- It is for opening and closing of a file which contains the paragraph orsentence.

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

5. 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 ischecked for its grammatical and overall correctness.

*Again here, we use try and except blocks for dealing with exceptional cases(i.e. the sentences orword which isn't there in our database.)

*It checks the sentence and then prompt the message accordingly.(i.e. it can even raise an errormessage)

6.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 terminalwindow.

*If error is received then the part of sentence or word is searched and then it send back to theprevious functions for re-checking and correction.

*Finally, the corrected sentence is recieved and then gets printed as the output.

Algorithms:

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

Testing:

We have successfully implemeted the program and got desired output.

Lets check for our grammer

```
# calling main function with sentences
main("english is not so that very eassy.")
```

```
ORIGINAL TEXT: english is not so that very eassy.
CORRECTED TEXT: English is not so that very easy.
```

```
[8] main('i am nnnot okka')
```

```
ORIGINAL TEXT: i am nnnot okka
CORRECTED TEXT: I am not okay
```

Lets check for our grammer

```
[14] # calling main function with sentences
     main("wer do you go to scul?")
```

```
ORIGINAL TEXT: wer do you go to scul?
CORRECTED TEXT: Where do you go to school?
```

```
main('good for people who are English learners.')
```

```
ORIGINAL TEXT: good for people who are English learners.
CORRECTED TEXT: Good for people who are English learners.
```

Lets check for our grammer

```
[21] # calling main function with sentences
     main("My coussin is very picky about the restaurants he dines in.")
```

```
ORIGINAL TEXT: My coussin is very picky about the restaurants he dines in.
CORRECTED TEXT: My cousin is very picky about the restaurants he dines in.
```

```
main('There are two very good reason for this.')
```

```
ORIGINAL TEXT: There are two very good reason for this.
CORRECTED TEXT: There are two very good reasons for this.
```

Lets check for our grammer

```
[28] # calling main function with sentences
     main(" The marble statue hed a big hed.")
```

```
ORIGINAL TEXT:  The marble statue hed a big hed.
CORRECTED TEXT:  The marble statue had a big head.
```

```
main('I want to rid a camel.')
```

```
ORIGINAL TEXT: I want to rid a camel.
CORRECTED TEXT: I want to ride a camel.
```

```
[43] # calling main function with sentences
     main("The bed room is comfortable.")
```

```
ORIGINAL TEXT: The bed room is comfortable.
CORRECTED TEXT: The bedroom is comfortable.
```

```
[44] main('The colour purple is my favorite.')
```

```
ORIGINAL TEXT: The colour purple is my favorite.
CORRECTED TEXT: The color purple is my favorite.
```

```
[45] main('My friend john is not well today.')
```

```
ORIGINAL TEXT: My friend john is not well today.
CORRECTED TEXT: My friend John is not well today.
```

```
main('I went to to the store.')
```

```
ORIGINAL TEXT: I went to to the store.
CORRECTED TEXT: I went to the store.
```

CHAPTER 5
RESULTS AND DISCUSSION

It first detects the grammatical errors and contextual errors from a given sentence or paragraphand then gives the corrected text.

Examples:

1. **Spelling :** It corrects a word which is not spelled correctly.

ORIGINAL TEXT: The marble statue had a big hed.

CORRECTED TEXT: The marble statue had a big head.

2. **Misused:** It corrects the confusion between words which sound similar or have a similarspelling.

ORIGINAL TEXT: I want to rid a camel.

CORRECTED TEXT: I want to ride a

camel.

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 comfortable.

CORRECTED TEXT: The bedroom is comfortable.

4. **Subject Verb Agreement :**It corrects the subject verb agreement.

ORIGINAL TEXT: The smell of flowers bring back memories.

CORRECTED TEXT: The smell of flowers brings back

memories.

5. **Common and Proper Nouns :** It corrects a proper or common noun which is notcapitalized.

ORIGINAL TEXT: My friend john is not well today.

CORRECTED TEXT: My friend John is not well

today.

6. **Double Words :** It corrects accidental repetition of a word.

ORIGINAL TEXT: I went to to the store.

CORRECTED TEXT: I went to the store.

7. **Double Words :** It corrects accidental repetition of a word.

ORIGINAL TEXT: I went to to the store.

CORRECTED TEXT: I went to the store.

8. **Indefinite Article:** It corrects confusion between a/an or omission of an indefinite articlewhen it is required .

ORIGINAL TEXT: John is studying for a MBA

degree. CORRECTED TEXT: John is studying for an

MBA degree.

9. **Definite Article:** It corrects omission of the definite article ("the") when it is required.

ORIGINAL TEXT: I had time of my life on this

vacation. CORRECTED TEXT: I had the time of my

life on this vacation.

10. **Consecutive Nouns :** It corrects wrong usage of two or more nouns in a row, either inpossessive form or as modifiers of each other.

ORIGINAL TEXT: Sheryl went to the tickets office.

CORRECTED TEXT: Sheryl went to the ticket

office.

11. **Plurality:** It corrects the confusion between the singular and plural form of a noun.

ORIGINAL TEXT: We bought a number of item.

CORRECTED TEXT: We bought a number of items.

12. **Pronouns:** It corrects the wrong pronoun or not using a pronoun when one is required.

ORIGINAL TEXT: I need you help.

CORRECTED TEXT: I need your help.

13. **Comparative Superlative:** It corrects the wrong usage of comparative and superlativestructures.

ORIGINAL TEXT: This movie is bad than anything I have seen.

CORRECTED TEXT: This movie is worse than anything I have seen.

14. **Tenses:** It corrects the wrong verb tense or form is being used.

ORIGINAL TEXT: It is important to submitted the paper on time. CORRECTED TEXT: It is important to submit the paper on time.

15. **Present Simple :** It corrects by applying the present simple tense.

ORIGINAL TEXT: The battery is lasting for only two hours.CORRECTED TEXT: The battery lasts for only two hours.

16. **Present Perfect:** It corrects by applying the present perfect tense.

ORIGINAL TEXT: I have develop this idea for days.

CORRECTED TEXT: I have developed this idea for days.

17. **Present Progressive:** It corrects by applying the present progressive tense.

ORIGINAL TEXT: Terry has writing a letter at the

moment. CORRECTED TEXT: Terry is writing a

letter at the moment.

18. **Past Simple:** It corrects applying the past simple tense.

ORIGINAL TEXT: I go to the store yesterday.

CORRECTED TEXT: I went to the store

yesterday.

19. **Past Perfect:** It corrects by applying the past perfect tense.

ORIGINAL TEXT: My parents has never been to Florida before this

summer. CORRECTED TEXT: My parents had never been to Florida

before this summer.

20. **Past Progressive:** It corrects by applying the past progressive tense.

ORIGINAL TEXT: He was sell fruit on the side of the road.

CORRECTED TEXT: He was selling fruit on the side of

the road.

21. **Future Tense:** It corrects by applying the future tense.

ORIGINAL TEXT: Amy try going to the chess club tomorrow.

CORRECTED TEXT: Amy will try going to the chess club

tomorrow.

22. **Subject Verb Agreement:** It corrects mismatch between the plurality of the

verb and thesubject.

ORIGINAL TEXT: A bouquet of yellow roses lend color and fragrance to the room.

CORRECTED TEXT: A bouquet of yellow roses lends color and fragrance to the room.

23. **The Infinitive:** It corrects using the infinitive form.

ORIGINAL TEXT: She expects it is ready by five.

CORRECTED TEXT: She expects it to be ready by

five.

24. **Participles:** It corrects errors related to incorrect usage or incorrect form of participles.

ORIGINAL TEXT: We are looking for employees with proved experience.

CORRECTED TEXT: We are looking for employees with proven

experience.

25. **Adverbial Modifiers:** It corrects incorrect use of adverbs instead of

adjectives or vice-versa or using the wrong adverb.

ORIGINAL TEXT: She is a beautifully woman.

CORRECTED TEXT: She is a beautiful woman.

26. **Prepositions:** It corrects the wrong preposition.

ORIGINAL TEXT: We arrived to the

station. CORRECTED TEXT: We arrived

at the station.

27. **Prepositions In/On/At creates confusion:** It corrects the confusion

between the usage ofthe prepositions in, on and at.

ORIGINAL TEXT: She lives in 666 Elm Street.

CORRECTED TEXT: She lives at 666 Elm Street.

ORIGINAL TEXT: The relevant data appeared on the rightmost

column. CORRECTED TEXT: The relevant data appeared in

the rightmost column.ORIGINAL TEXT: They laughed on all

the jokes.

CORRECTED TEXT: They laughed at all the jokes.

28. **Beginning Of Sentence Capitalization:** It corrects the first word in a sentence which isnot capitalized.

ORIGINAL TEXT: this is not right.

CORRECTED TEXT: This is not right.

29. **Comma Addition:** It adds a comma where there should be one, e.g. after an introductoryphrase, between list items and so on.

ORIGINAL TEXT: Students will demonstrate understanding of spoken words syllables andsounds.

CORRECTED TEXT: Students will demonstrate understanding of spoken words, syllables andsounds.

30. **Question Mark Addition:** It corrects by adding a question mark at the end of a sentence .

ORIGINAL TEXT: How did you manage to do it

CORRECTED TEXT: How did you manage to do it?

31. **Vocabulary:** It corrects semantically unsuitable word in a sentence.

ORIGINAL TEXT: Can you make me a

favor? CORRECTED TEXT: Can you do

me a favor?

32. **Singular/Pural nouns:** It corrects singular and pural nouns.

ORIGINAL TEXT: Six people lost their life in the

accident. ORIGINAL TEXT: Six people lost their

lives in the accident.

33. **Consecutive Nouns:** It corrects consecutive nouns.

ORIGINAL TEXT: Sheryl went to the tickets

office. CORRECTED TEXT: Sheryl went to

the ticket office.

34. **Misused Words Correction:** It corrects misused words.

Using its contextual grammar checker, Ginger recognizes the misused words in any sentence andreplaces them with the correct ones.

ORIGINAL TEXT: I was wandering if there's any

news. CORRECTED TEXT: I was wondering if

there's any news.

35. **Contexual Spelling Correction:** It corrects contextual spelling.

The Grammar Checker is a contextual spell checker which identifies the correction that best fits the meaning of the original sentence. When combined with the Grammar Checker, you can correct entire sentences in a single click. The same misused word will have a different correctionbased on the context.

ORIGINAL TEXT: The marble statue hed a big hed .

CORRECTED TEXT: The marble statue had a big head.

35) **Phonetic spelling:** It corrects phonetic spelling.

Phonetic spelling mistakes are corrected even if the correct spelling is very different from the waythey were originally written.

ORIGINAL TEXT: I like books, exspecaley the

classics. CORRECTED TEXT: I like books,

especially the classics.

36) **Irregular verb conjugations:** It corrects irregular verb conjugations.

Irregular verb conjugations are corrected

as well.ORIGINAL TEXT: He flyed to

Vancouver.

CORRECTED TEXT: He flew to Vancouver.

CHAPTER 6

CONCLUSION AND FUTURE WORK

I have successfully completed the project.

Link to project:

https://colab.research.google.com/drive/1VTLBDX0qYa8wOGDXjZLBjbVZPo8567x3?usp=shar ing

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.Sowe need to correct the grammar for other languages.

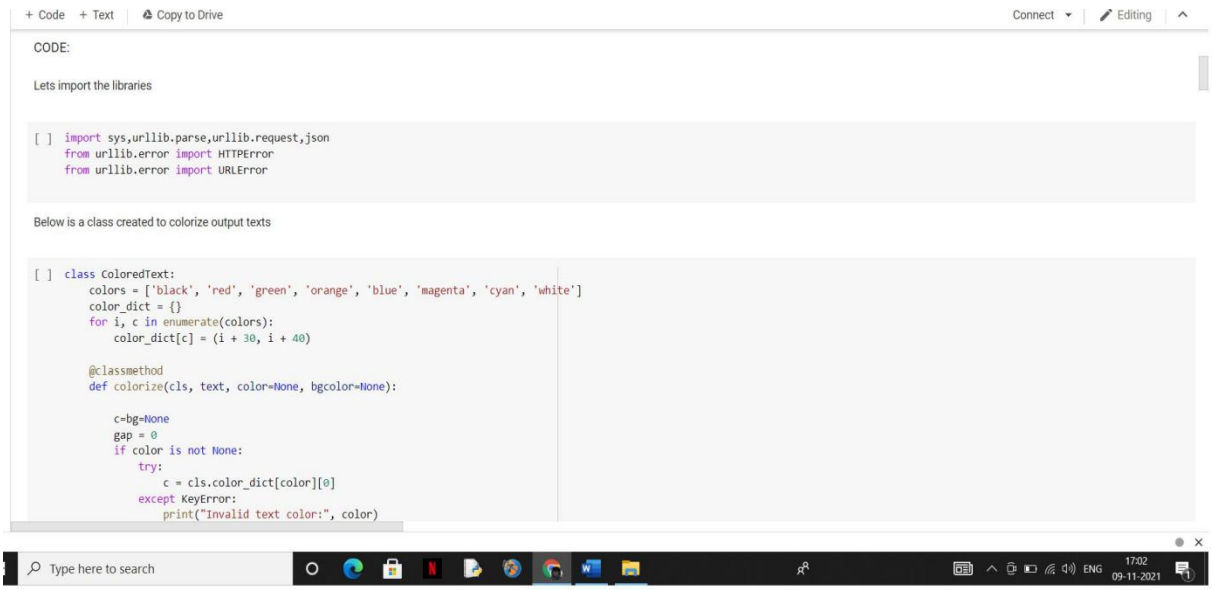
4. We need to add a feature named "Numeral Spelling Out" which write digits instead ofspelling out the numbers 0-9.

5. The sentence should be written in utf-8 format so we need to add other writing format.

CHAPTER 7

APPENDICES

SOURCE CODE



+ Code + Text | Copy to Drive — Connect | Editing

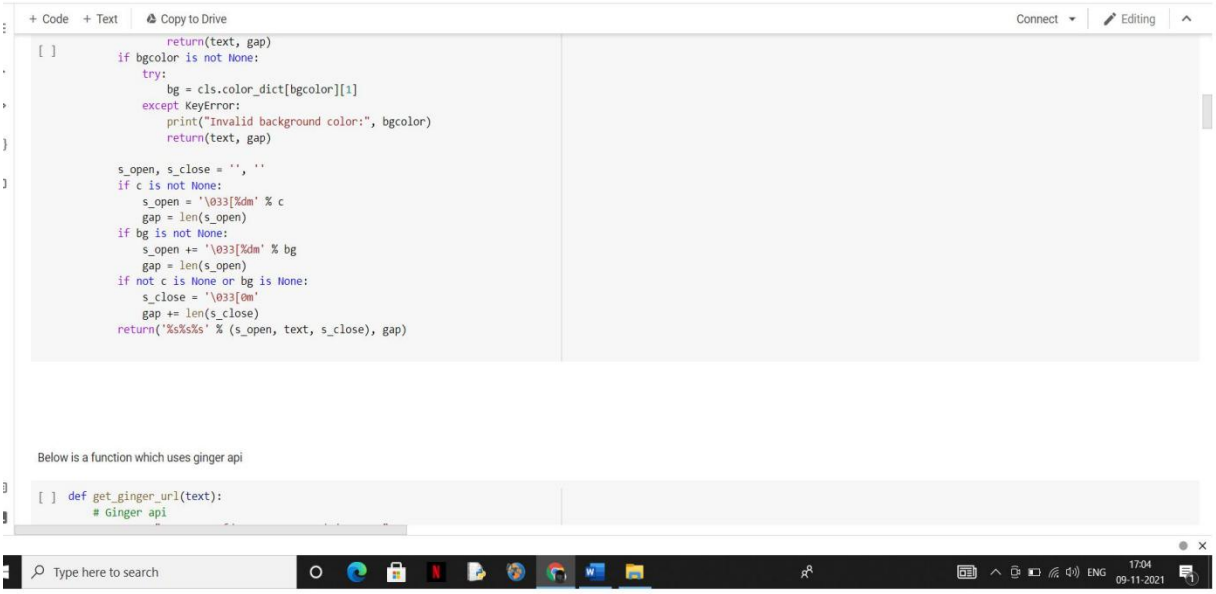
CODE:

Lets import the libraries

```
[ ] import sys,urllib.parse,urllib.request,json
    from urllib.error import HTTPError
    from urllib.error import URLError
```

Below is a class created to colorize output texts

```
[ ] class ColoredText:
        colors = ['black', 'red', 'green', 'orange', 'blue', 'magenta', 'cyan', 'white']
        color_dict = {}
        for i, c in enumerate(colors):
            color_dict[c] = (i + 30, i + 40)

        @classmethod
        def colorize(cls, text, color=None, bgcolor=None):

            c=bg=None
            gap = 0
            if color is not None:
                try:
                    c = cls.color_dict[color][0]
                except KeyError:
                    print("Invalid text color:", color)
```

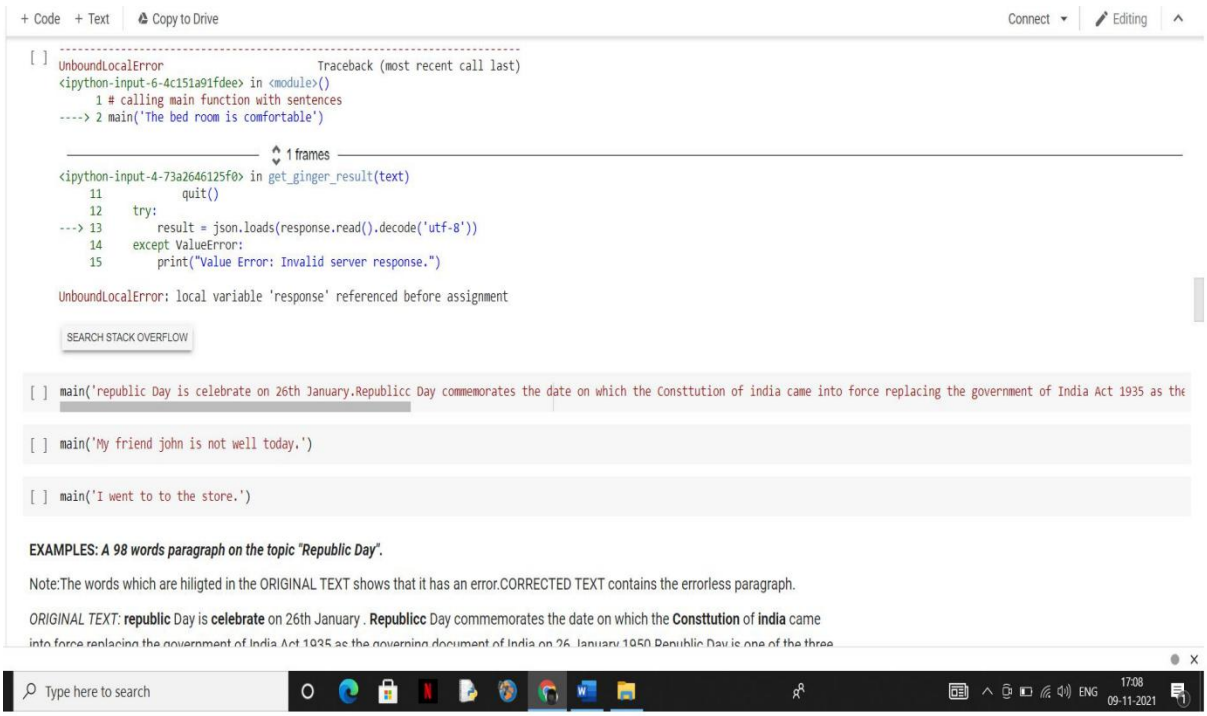



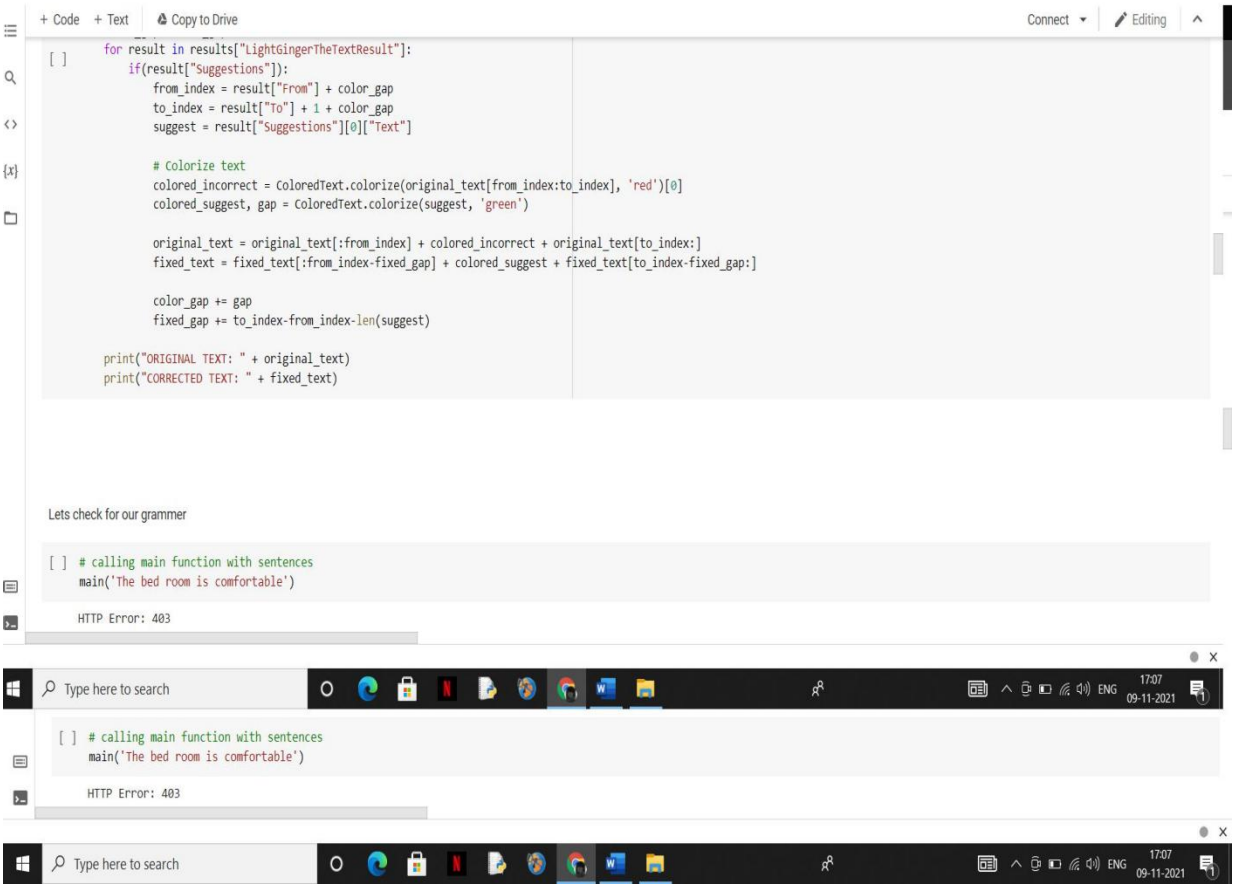
+ Code + Text | Copy to Drive — Connect | Editing

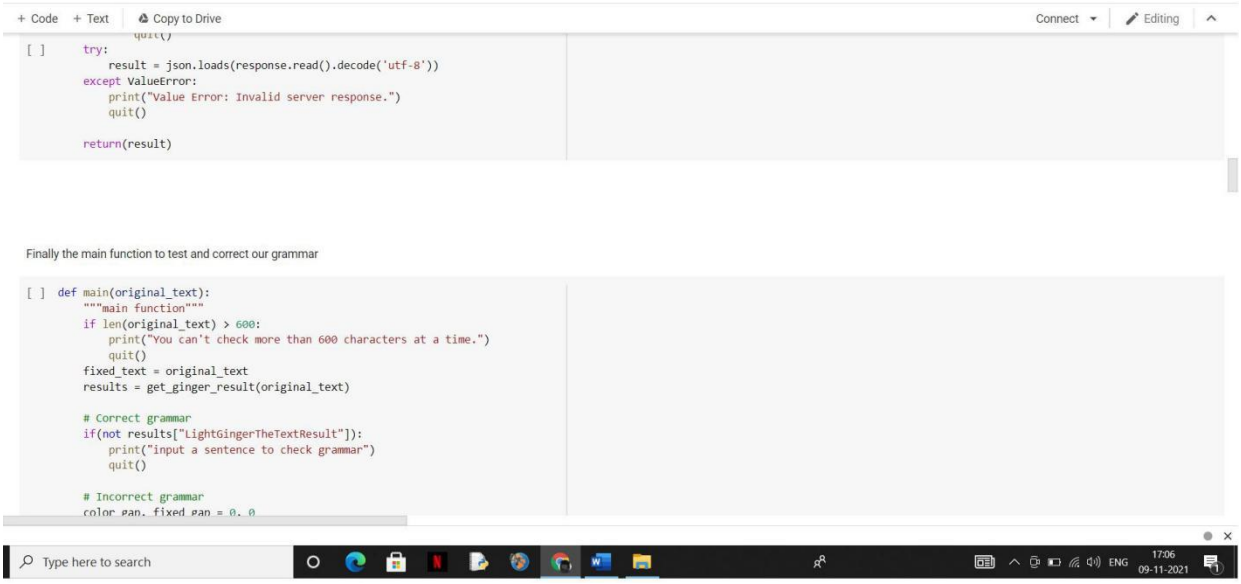
```
[ ]             return(text, gap)
            if bgcolor is not None:
                try:
                    bg = cls.color_dict[bgcolor][1]
                except KeyError:
                    print("Invalid background color:", bgcolor)
                    return(text, gap)

            s_open, s_close = '', ''
            if c is not None:
                s_open = '\033[%dm' % c
                gap = len(s_open)
            if bg is not None:
                s_open += '\033[%dm' % bg
                gap = len(s_open)
            if not c is None or bg is not None:
                s_close = '\033[0m'
                gap += len(s_close)
            return('%s%s%s' % (s_open, text, s_close), gap)
```

Below is a function which uses ginger api

```
[ ] def get_ginger_url(text):
        # Ginger api
```


+ Code + Text Copy to Drive
Connect Editing
UnboundLocalError Traceback (most recent call last)
<ipython-input-6-4c151a91fdee> in <module>()
1 # calling main function with sentences
----> 2 main('The bed room is comfortable')
1 frames
<ipython-input-4-73a2646125f0> in get_ginger_result(text)
11 quit()
12 try:
---> 13 result = json.loads(response.read().decode('utf-8'))
14 except ValueError:
15 print("Value Error: Invalid server response.")
UnboundLocalError: local variable 'response' referenced before assignment
SEARCH STACK OVERFLOW
main('republic Day is celebrate on 26th January.Republicc Day commemorates the date on which the Consttution of india came into force replacing the government of India Act 1935 as the
main('My friend john is not well today.')
main('I went to to the store.')
EXAMPLES: A 98 words paragraph on the topic "Republic Day".
Note:The words which are hiligted in the ORIGINAL TEXT shows that it has an error.CORRECTED TEXT contains the errorless paragraph.
ORIGINAL TEXT: republic Day is celebrate on 26th January . Republicc Day commemorates the date on which the Consttution of india came
into force replacing the government of India Act 1935 as the governing document of India on 26 January 1950 Republic Day is one of the three
Type here to search
ENG
17:08
09-11-2021


+ Code + Text Copy to Drive
Connect Editing
for result in results["LightGingerTheTextResult"]:
if(result["Suggestions"]):
from_index = result["From"] + color_gap
to_index = result["To"] + 1 + color_gap
suggest = result["Suggestions"][0]["Text"]
# Colorize text
colored_incorrect = ColoredText.colorize(original_text[from_index:to_index], 'red')[0]
colored_suggest, gap = ColoredText.colorize(suggest, 'green')
original_text = original_text[:from_index] + colored_incorrect + original_text[to_index:]
fixed_text = fixed_text[:from_index-fixed_gap] + colored_suggest + fixed_text[to_index-fixed_gap:]
color_gap += gap
fixed_gap += to_index-from_index-len(suggest)
print("ORIGINAL TEXT: " + original_text)
print("CORRECTED TEXT: " + fixed_text)
Lets check for our grammer
# calling main function with sentences
main('The bed room is comfortable')
HTTP Error: 403
Type here to search
ENG
17:07
09-11-2021
# calling main function with sentences
main('The bed room is comfortable')
HTTP Error: 403
Type here to search
ENG
17:07
09-11-2021

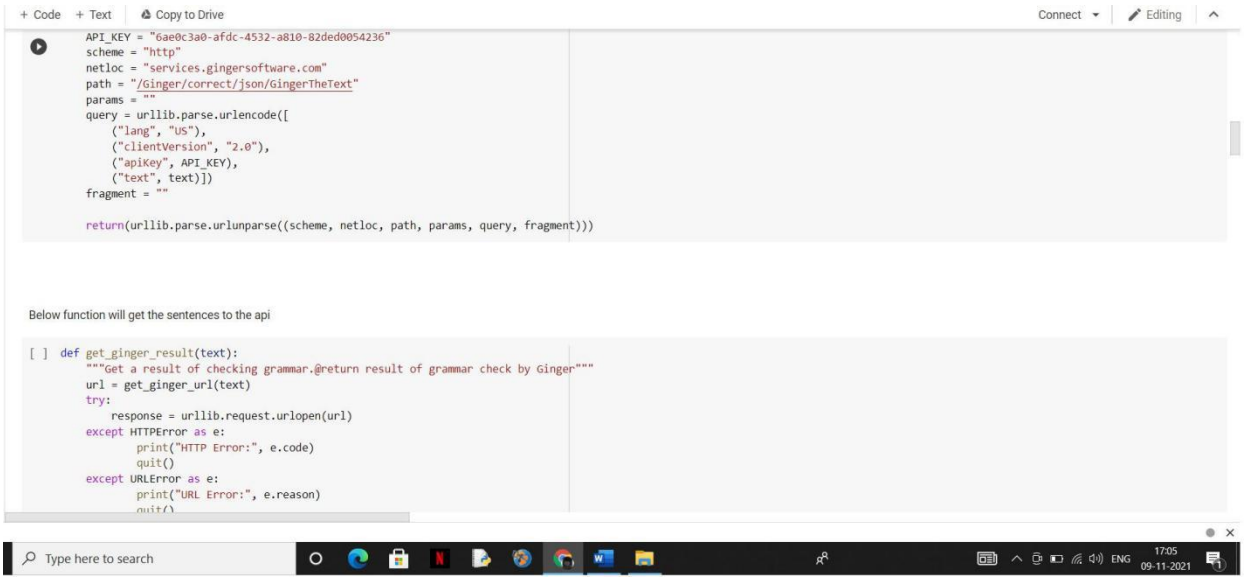


```
    try:
        result = json.loads(response.read().decode('utf-8'))
    except ValueError:
        print("Value Error: Invalid server response.")
        quit()

    return(result)
```

Finally the main function to test and correct our grammar

```
def main(original_text):
    """main function"""
    if len(original_text) > 600:
        print("You can't check more than 600 characters at a time.")
        quit()
    fixed_text = original_text
    results = get_ginger_result(original_text)

    # Correct grammar
    if(not results["LightGingerTheTextResult"]):
        print("input a sentence to check grammar")
        quit()

    # Incorrect grammar
    color_gap, fixed_gap = 0, 0
```


LEL



```
    API_KEY = "6ae0c3a0-afdc-4532-a810-82ded0054236"
    scheme = "http"
    netloc = "services.gingersoftware.com"
    path = "/Ginger/correct/json/GingerTheText"
    params = ""
    query = urllib.parse.urlencode([
        ("lang", "US"),
        ("clientVersion", "2.0"),
        ("apiKey", API_KEY),
        ("text", text)])
    fragment = ""

    return(urllib.parse.urlunparse((scheme, netloc, path, params, query, fragment)))
```

Below function will get the sentences to the api

```
def get_ginger_result(text):
    """Get a result of checking grammar.@return result of grammar check by Ginger"""
    url = get_ginger_url(text)
    try:
        response = urllib.request.urlopen(url)
    except HTTPError as e:
            print("HTTP Error:", e.code)
            quit()
    except URLError as e:
            print("URL Error:", e.reason)
            quit()
```

LEARNING EXPERIENCES

Learnt many new things about machine learning and came to know about the vast field pfmachine learning. It was great to learn.

I came to know new things about ML and python programming language , learnt new things about Gibger API and how API works. Overall it was great experience , Thanks to my Project Guide Dr S Jerald Nirmal Kr. Sir who helped and guided us a lot.
Thanks to my project partner for his contribution and great efforts.

CHAPTER 8

REFRENCES

1. https://software.intel.com/en-us/ai/courses/artificial-intelligence
2. https://www.scipy.org/docs.html
3. https://www.tensorflow.org/tutorials/keras/classification
4. https://software.intel.com/en-us/ai/courses/deep-learning
5. https://software.intel.com/en-us/ai/courses/machine-learning
6. https://scikit-learn.org/stable/tutorial/index.html
7. https://www.tensorflow.org/tutorials/images/cnn
8. https://www.python-course.eu/machine_learning.php