TAGMEMIC- A LINGUISTIC UNIT

Submitted in partial fulfilment of the requirements

of the degree of

Bachelor of Engineering in Information Technology

By

PRATYUSHA TRIVEDI (Roll No. 16103B0011)
ALIFYA KHAN (Roll No. 16101A0065)
KARTHIK ASHOK (Roll No. 16101A0004)

Under the Guidance of

Prof. KANCHAN DHURI

Department of Information Technology

VIT | Vidyalankar Institute of Technology
ACCREDITED A+ BY NAAC

Vidyalankar Institute of Technology

Wadala(E), Mumbai-400437

University of Mumbai

2019-20

CERTIFICATE OF APPROVAL

This is to certify that the project entitled

“Tagmemic- A linguistic Unit”

is a bonafide work of

PRATYUSHA TRIVEDI (Roll No. 16103B0011)
ALIFYA KHAN (Roll No. 16101A0065)
KARTHIK ASHOK (Roll No. 16101A0004)

submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the

degree of

Undergraduate in **“INFORMATION TECHNOLOGY”**.

Guide	Head of Department	Principal
(Prof. Kanchan Dhuri)	(Dr. Deepali Vora)	(Dr. S.A. Patekar)

Project Report Approval for B. E.

This project report entitled ***Tagmemic- A Linguistic Unit*** by

1. Pratyusha Trivedi (16103B0011)
2. Alifya Khan (16101A0065)
3. Karthik Ashok (16101A0004)

is approved for the degree of ***Bachelor of Engineering in Information Technology.***

Examiners

1.----------

2.----------

Date:

Place:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name of student	Roll No.	Signature
1. Pratyusha Trivedi	16103B0011	
2. Alifya Khan	16101A0065	
3. Karthik Ashok	16101A0004	

Date:

ACKNOWLEDGEMNT

We are honoured to present "**Tagmemic- A Linguistic Unit**" as our B.E Final Year Project. We are using this opportunity to express our profound gratitude to our principal "**Dr. Sunil Patekar**" for providing us with all proper facilities and support.

We express our deepest gratitude towards our HOD "**Dr. Deepali Vora**" for her valuable and timely advice during the various phases in our project. We would like to thank our project guide **"Prof. Kanchan Dhuri"** for support, patience and faith in our capabilities and for giving us flexibility in terms of working and reporting schedules. Finally, we would like to thank everyone who have helped us directly or indirectly in our project.

We would also like to thank our staff members and lab assistant for permitting us to use computer in the lab as when required. We thank our college for providing us with excellent facilities that helped us to complete and present this project.

Pratyusha Trivedi

Alifya Khan

Karthik Ashok

ABSTRACT

In the current scenario, there are different apps, websites, etc. to carry out different functionalities with respect to text such as grammar correction, translation of text, extraction of text from image or videos, etc. There is no app or a website where a user can get all these functions/features at one place and hence the user is forced to install different apps or visit different websites to carryout those functions. The proposed system identifies this problem andtries to overcome it by providing various text-based features at one place, so that the user will not have to hop from app to app or website to website to carry out various functions. The proposed system will provide users with various functions such as grammar checking, text extraction from an image, text summarization, translation of text into different languages, etc. which will help them in their daily life.

CONTENTS

List of Figures

CHAPTER 1

INTRODUCTION

INTRODUCTION

Every individual today leads a fast life. Even if the most common tasks are made convenient, it contributes largely to saving one's time, money and energy. Right from corporate staff and businessmen to students and senior citizens everybody is bound to use text related applications. Most commonly people have to write different types of documents and face the problem of correct use of English grammar. Reading long lines of articles might be stressful and time-consuming. Similarly, not being able to understand a document due to the language barrier may lead to a delay in work. Also, being able to use the text in an image for any of the above purposes is not possible under a single frame. Hence, this system identifies the above problems and provides a solution in an all in one web-page. With the help of this system, one can easily navigate through different functionalities as per their use. The system is largely divided into four modules. The first one deals with extraction of text from an image which provides the functionality of obtaining the text within any image. The second module helps in summarization of any article to desired number of lines as per user input. Further, the third module deals with translation of English text to Spanish, German, Hindi and French as these are the most widely used languages around the world. Finally, the fourth module helps the user in correcting the grammatical errors within a sentence.

1.1 Problem Statement

Simple tasks like reading a hoarding may be a stressful one if written in a different language. One has to note down the entire text written in the hoarding and then translate it via a different application. This might be time-consuming and not feasible in every instance. Further, various documents demand different styles of writing and the use of correct grammar is one of the major problems while doing this. This does not let the user work according to their comfort which leads in delay and inefficiency. Also, reading long lines of articles may not be feasible in many instances due to time constraints. These different modules are the most basic and useful functionalities one uses in their day to day life and hence it is important to identify them. This system has been developed by considering all the above problems and understanding the user's need and purpose.

.

1.2 Scope

This system aims at providing a single platform for the text functionalities such as Text extraction from an image, Text translation, Text summarization and Grammar check. A website is provided where all these functionalities are available for easy and efficient use of all the modules.

1.3 Motivation

Most commonly while working on any type of document the writer faces problems like incorrect grammar usage which leads to poor presentation of the document. Similarly, a high count of lines in a document tends to miss the core point of the document. Sometimes, a document in a specific language needs to be translated to another language due to language barrier. Failing to do so may make the user confused and unable to read the document. Further, if the user wants to obtain the text from an image for their own use, this is not possible in one go. Hence, by identifying these issues a system is developed where all such similar functionalities are provided under a single platform.

The first module focuses on extracting text from any kind of image and stores it in a document for usage. The second module deals with translation of text from English to Spanish, Hindi, French and German. As these are the most widely spoken languages, they are chosen for this system. The third module is based on correcting grammatical errors in English. Once user inputs a sentence, grammatical errors are checked along with the spelling errors and a list of suggestions are provided to the user with respect to the incorrectly spelled word. The corrected sentence is thus provided as output for further use. The final module is summarisation of an article to specified number of sentences as mentioned by the user. All these modules in tandem complete this system and provide easy to use web interface for the user.

CHAPTER 2

LITERATURE SURVEY

LITERATURE SURVEY

2.1 Survey Based on Research Papers

The method used by Satish Kumar, Sunil Kumar and Dr. S. Gopinath in their research paper to extract the text from an image follows five steps. [1] The first step is to convert the coloured image to grey scale image. Here, the coloured images which can follow the RGB or CYM pattern which include information of each of the coloured pixel is converted to grey scale images where the darkest black can be 00000000 and lightest white can be 11111111 respectively. In the second step, binarization is applied where pixels higher than the threshold value are changed to 0 and pixels less than the threshold value are changed to 1. In the third and fourth steps, the connected components via neighbour distance is checked and the transition of pixels horizonatlly and search vertically for finding the area of text.

In the case of translation of text, the research paper published by Sandeep Saini, Vineet Sahula suggests the different techniques that are available for machine translation (MT). [2] Like human translators, machine translators also have four kinds of memories. But, the cognitive approach proves to be the most difficult in developing MT systems. However, the three main approaches are Hybird, Rule-based and Corpus-based translation. Rule-based approach is easy to implement. The input text is parsed and the corresponding text is obtained. However, the corpus-based translation requires large data sets and is based on statistical analysis which explains that every sentence in a particular language can be converted to any language in many different ways.

N.Moratanch ,S.Chitrakala mentions in their research paper that summarization is the most useful and can be performed in many different ways. [3] Broadly, there are two major ways, unsupervised learning, and supervised learning. Unsupervised learning has four approaches. The first one focuses on graph-based approach. The sentences are assigned as a graph and the

edges are assigned cosine values. Based on these values the LexRank scores are determined. The second approach is based on fuzzy logic. Length of the sentence, the similarity of sentence are the inputs to the fuzzy system. Further, feature extraction is performed and the ranks of sentences are determined on the basis of how the appear in the original document. The third approach which is the concept-based approach is based on getting the text from an outer knowledge base. A vector model is designed which determines the similarity and corresponding summarization. The last approach called the Latent Semantic Analysis does not require an external knowledge base. It counts the frequency of words within a sentence and the sentences producing the highest frequency of words are selected for summarization. Further, the supervised learning approach deals with machine learning approach which follows the Bayes rule.

As far as the correction of grammar is concerned, Vibhakti V. Bhaire, Ashiki A. Jadhav, Pradnya A. Pashte, Mr. Magdum P.G state in their paper that it is very important to first correct the spellings of words within a sentence. [4] This is achieved by first identifying and comparing each word with the words already present in the database. Any incorrectly spelled word is replaced with the corresponding word in the database. The database also contains lexical information and grammatical rules. Further, by splitting the sentence, identifying the part of speech and comparing it with the general Subject+Verb+Object pattern the words are re-arranged and the sentence is corrected.

The Django framework is one of fastest frameworks nowadays. It is quick, secure, flexible and is a benefit to both developers and users at the same time. In the research paper published by Prof. B Nithya Ramesh, Aashay R Amballi and Vivekananda Mahanta they suggest the steps for successfully using the Django framework. [5] The first step is to install Django. Every web-based framework uses the MVC concept. M stands for "Model" which helps us get data from the source. V stands for "Views" which as the name suggests is the presentation part. Finally, C stands for "Controller" which communicates with the Model and View respectively.

2.2 Survey Based on Existing Applications/Websites

1. Text Scanner [OCR]

- The app supports over 50 languages including Chinese, Japanese, French, and more.
- Supports extracting text from handwritten text.
- The app interface features essential scanning functionalities like magnification and a brightness slider to capture text in the clearest way possible.

1. Heming Way

- Focuses on increasing the readability of the post.
- Highlights the phrases with different colours.
- The tool also denotes the passive voice from the sentences.
- The tool tells you which sentence is hard to read, and also it provides the suggestions to simplify the words.

2. OnlineCorrection

- Minimalistic Design.
- Check Grammar and Spelling Errors.
- Check Stylistic Issues.
- Stylistic Hints.
- Auto Correction.
- Suggestions For Sentence Construction and Vocabulary.
- Reporting Feature.
- Supports English Dialects.

4. Grammarly

- Detects potential grammar, spelling, punctuation, word choice, and style mistakes in writing.
- It's algorithms flag potential issues in the text and suggest context-specific.
- corrections for grammar, spelling, wordiness, style, punctuation, and plagiarism.
- It is available via a browser extension.

5. Reverso

- Instant translation in 11 languages: Spanish, French, Italian, English, Portuguese, German, Polish, Dutch, Arabic, Russian and Hebrew.
- Vocabulary lists based on a user's personal selection of examples and translations.
- Option to view reverse translations, definitions or conjugation when applicable.
- Flashcards to help memorize the words or phrases searched.

6. Ginger

- Ginger detects the errors very promptly and highlights them.
- When you hover the mouse over the highlighted word, then it will provide the suggestion for correcting it.
- Ginger, you can also either select British or the American English.
- Ginger also translate your document into 40 languages including Chinese, French, Urdu, Hindi, Arabic and Russian.
- One can search the definition of your words through the dictionary and rephrase your sentences for the variety of structure.
- One can also create a personal dictionary so that the program did not show errors for that words next time.
- Text reader is another unique Ginger feature which is useful for learning.

CHAPTER 3

SYSTEM DESIGN

PROPOSED SYSTEM

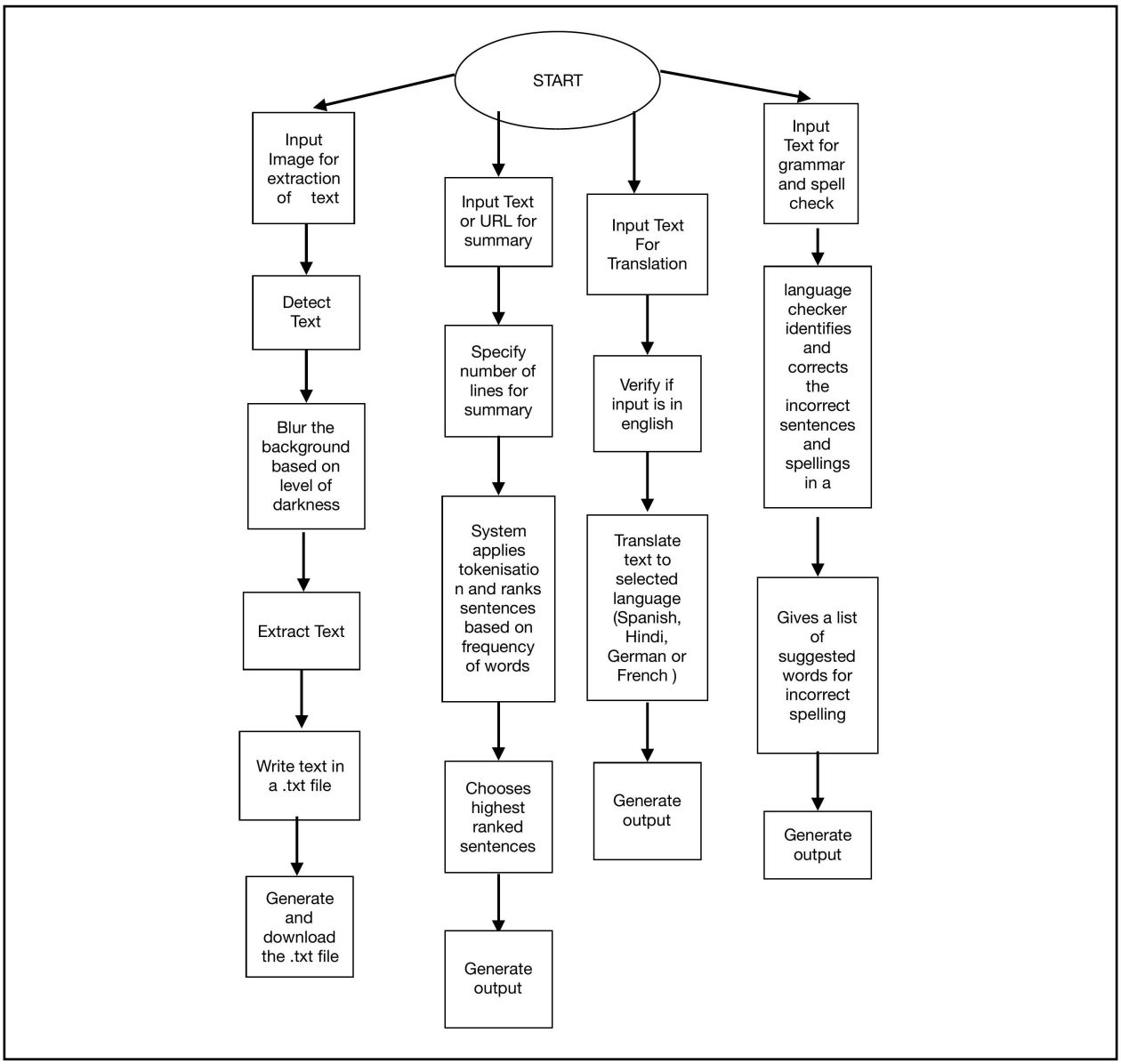

Fig 3.1: Flowchart

Flowchart Description:

- For the first module, the the image is first loaded into the system.
- The text in the image is detected and pre-processing techniques are applied where blurring is applied based on the darkness of the image input.
- Thus the text is extracted and written in a.txt file for output generation and is available to download.

- For the second module, the user first enters the text or url for summary.
- After specifying number of lines, the sentences are tokenized and the highest frequency words are ranked accordingly and chosen for the output.
- For the third module, once the user enters the text, it is verified whether the input is in English.
- By Python's translator library, the text is converted to Spanish, Hindi, French or German.
- For the final module, the language checker checks for the grammatical errors in a sentence.
- It provides a list of suggestions for the incorrect spelling entered and generates the output.

3.1 Proposed System

In the traditional system the user has to navigate from one application to the other which can be stressful and time consuming.

This system identifies the problem and provides an all in one platform where these functionalities will be available. This system is built using Python and HTML, CSS for developing the website. Since, the modules are written in Python and the webpage is based on HTML, Django is used to integrate them together.

The final output shows us how efficiently this system works and user convenience is achieved.

3.2 Methodology

In order to provide and solve the problem mentioned in our problem statement, this system is developed by considering and meeting the needs of the user. All the different modules such as Extraction of text from an image, Text translation, Text Summarization and Grammar Check are developed in Python.

This forms the base of our system. Further, for user interfacing purposes a webpage is designed and developed in HTML, CSS. However, as the functionalities are coded in Python, integrating them with the HTML based webpage is necessary and the most vital step. This is achieved by using Django framework which is throughly explained in the upcoming chapters.

3.3 Analysis

3.3.1 Process Model

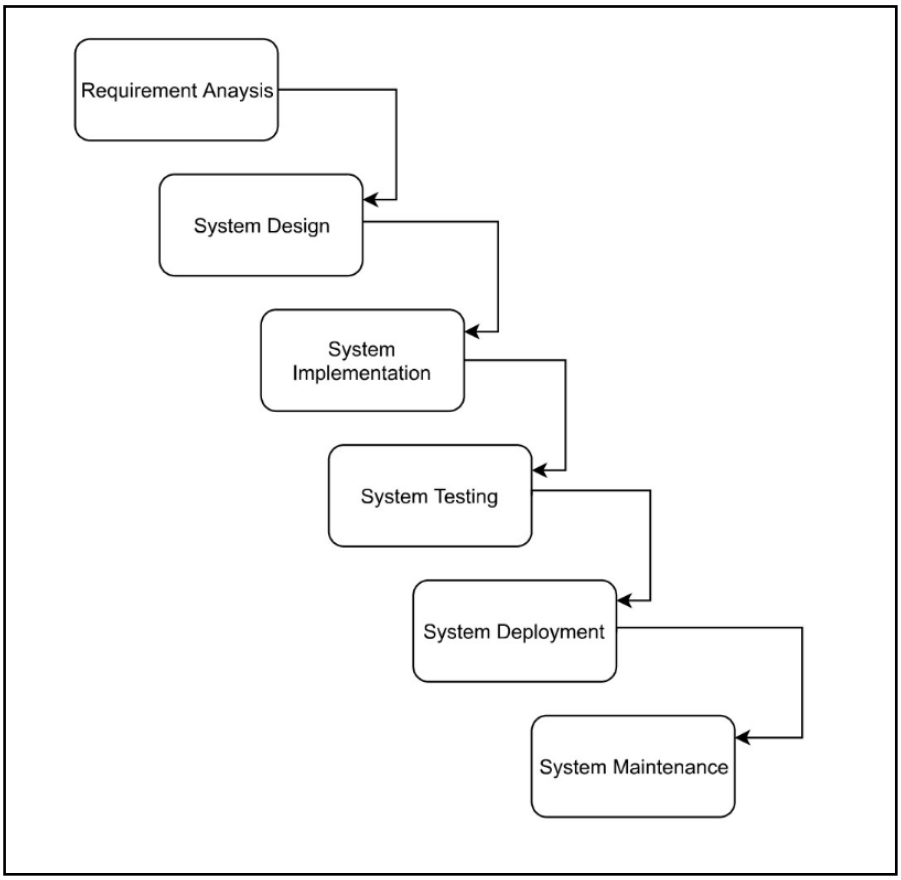

Fig 3.2: Waterfall Model

We will be using waterfall model in order to develop our project. The reasons for using waterfall model are as follows:

- It allows us to modulate and control it according to our requirement.
- It allows for efficient planning and development of project.
- A proper schedule can be set with deadlines for every stage of development.
- We have all the Requirements Ready for the proposed system.

3.3.2 Feasibility Analysis

1. **Technical feasibility**: Technical feasibility focuses on the technical resources (software and hardware) available for the project. It also helps to determine whether the team is capable of converting the ideas into working systems. The software required for the system are a basic text editor like Atom, Sublime Text for writing the code and using Python as the programming language. The hardware components are not required for this system.

2. **Economic feasibility**: This assessment typically involves a cost/ benefits analysis of the project. This project developed in a software based environment and hence the budget of this project is null.

3. **Operational feasibility**: This assessment involves a study to analyse and determine how well the organisation's needs can be met by completing the project. The main objective of the project is to firstly detect text from an image, translate a given text in English language to any specified language as per the user. Secondly, correct grammatical errors of any text in English language and summarisation of an article to reduced number of lines and increase reader's understanding and speed. As all these technicalities mentioned above are present in one system, it is very useful not only for students but for teachers and working professionals as well.

3.3.3 Timeline Chart

Task Name	Start	End	Duration (days)
Project Selection	28/2/19	5/3/19	6
Project Planning	8/3/19	12/3/19	5
Text Detection from Image	15/3/19	10/4/19	27
Text Extraction	June 2nd week	June 2nd week	7
Translation	June 4th Week	July 1st week	15
Paragraph Summarization	July 2nd Week	July 4th week	20
Grammar Check	July 1st Week	September 2nd week	65
Merging	November 2nd Week	December 2rd week	30
Web Application	December 3rd Week	January 4th Week	40

Fig 3.3: Timeline Chart

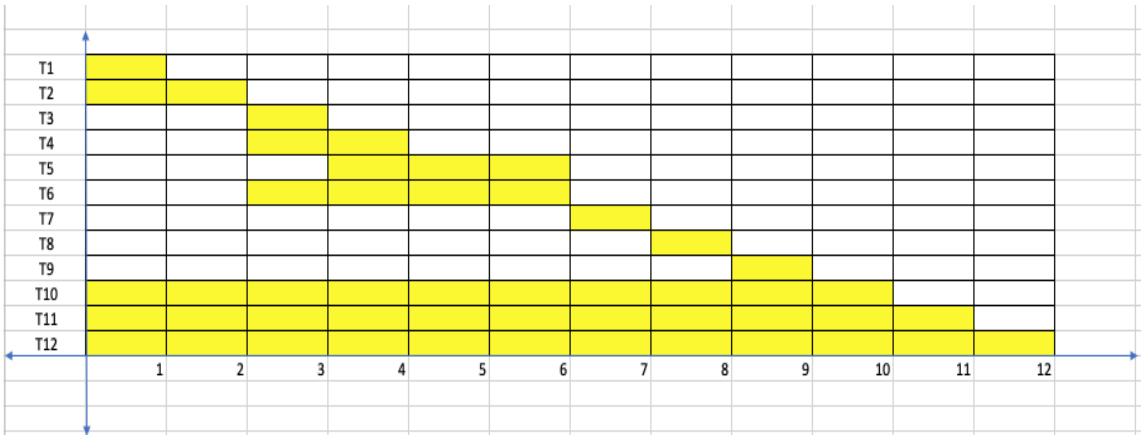

Fig 3.4: Gant Chart

CHAPTER 4

SYSTEM IMPLEMENTATION

SYSTEM IMPLEMENTATION

4.1 Module- wise Implementation

- **Text Extraction from An Image**

This feature extracts text out of an image. The open-source computer vision and tesseract which is an optical character recognition (OCR) engine for various operating systems are used to detect and identify text within an image. Computer vision sets up a pipeline between the logic code and capturing the vision. The main feature is that it is cross- platform. OpenCV is used for capturing the image for extraction of text and tesseract helps in converting the image into a machine-encoded script and provides the output. OCR is a tool used to electronically convert images to machine-encoded text. It is formally used to recognize text present in the images, followed by extracting the text and converting it into a machine-readable format. This system asks the user to choose and upload an image. The input image which is given from the user is loaded from disk in PIL format, a requirement when using tesseract. The image is loaded from the disk into memory followed by converting it to grey scale. By using pytesseract.image to string we convert the contents into our desired string then we pass a reference to the temporary page file residing on disk and then delete the temporary file. The extracted text is then printed in a formattable .txt file which is provided to the user to download.

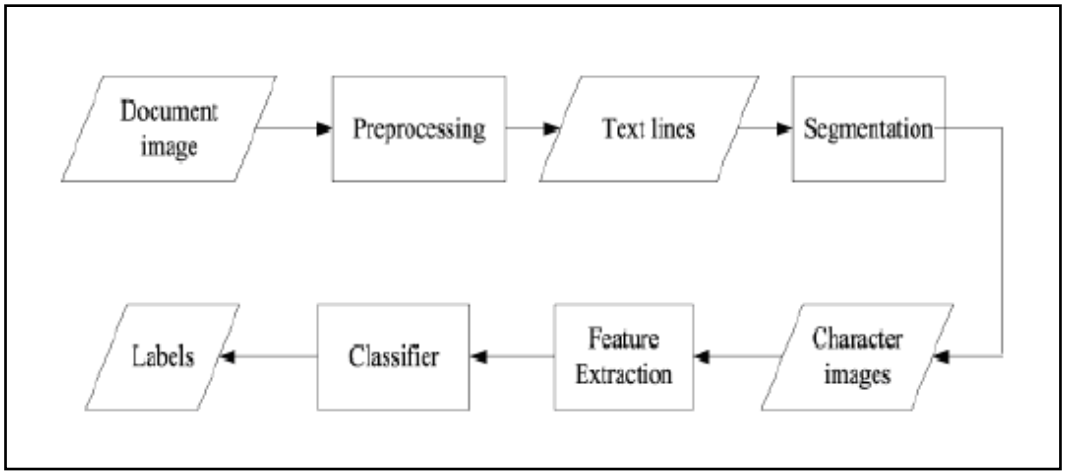

Fig 4.1: OCR Flow Diagram

- **Text Summarization**

This module focuses on shortening the length of an article while maintaining the gist of the article and providing meaningful sentences. Using NLTK libraries prove to be the most convenient. It is a set of libraries for Natural Language Processing for English written in Python. It gets the data ready by cleaning the text for applying machine learning and deep learning algorithms. Similarly, NLP is nothing but understanding human languages. NLTK is used in summarization for tokenizing and ranking the sentences respectively. The user will be prompted to enter the number of lines and based on the frequency of words used in a particular sentence, ranks are determined and the highest ranked sentences are chosen which provide the output successfully.

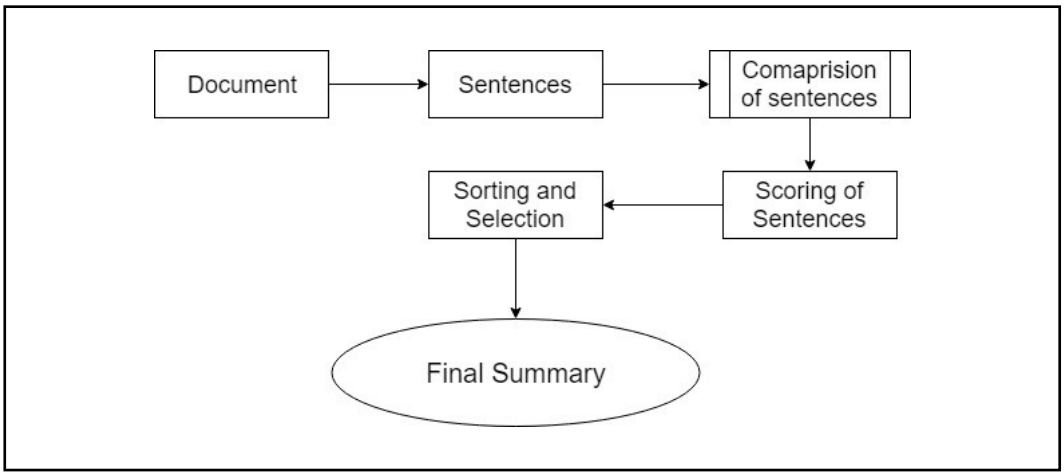

Fig 4.2: Text Summarization Flow Diagram

- **Grammar Check**

This function helps the user to correct the grammatical errors as well as spelling mistakes in a sentence. The spell checking is done by importing the pyspellchecker library in Python. It uses the Levenshtein Distance algorithm to find permutations of the misspelled words. Once the list of permutations is ready, it displays all the suggestions which are close to the incorrectly spelled word in the sentence. By comparing the permutations which include insertions, replacements, deletions, and transpositions of known words in the word frequency list. Those words that occur often in the frequency list are likely to be correct. For grammar check, the language-check library of Python is used. It tokenizes the sentences and breaks down the sentences into smaller sentences to identify the parts of speech. Parts of speech can be referred to as the syntax in any sentence.

These parts are compared with generalized patterns and the sentence is formatted accordingly. Hence, in this system, the spelling of the words and the structure of the sentence is checked, and the correct sentences are given as the output.

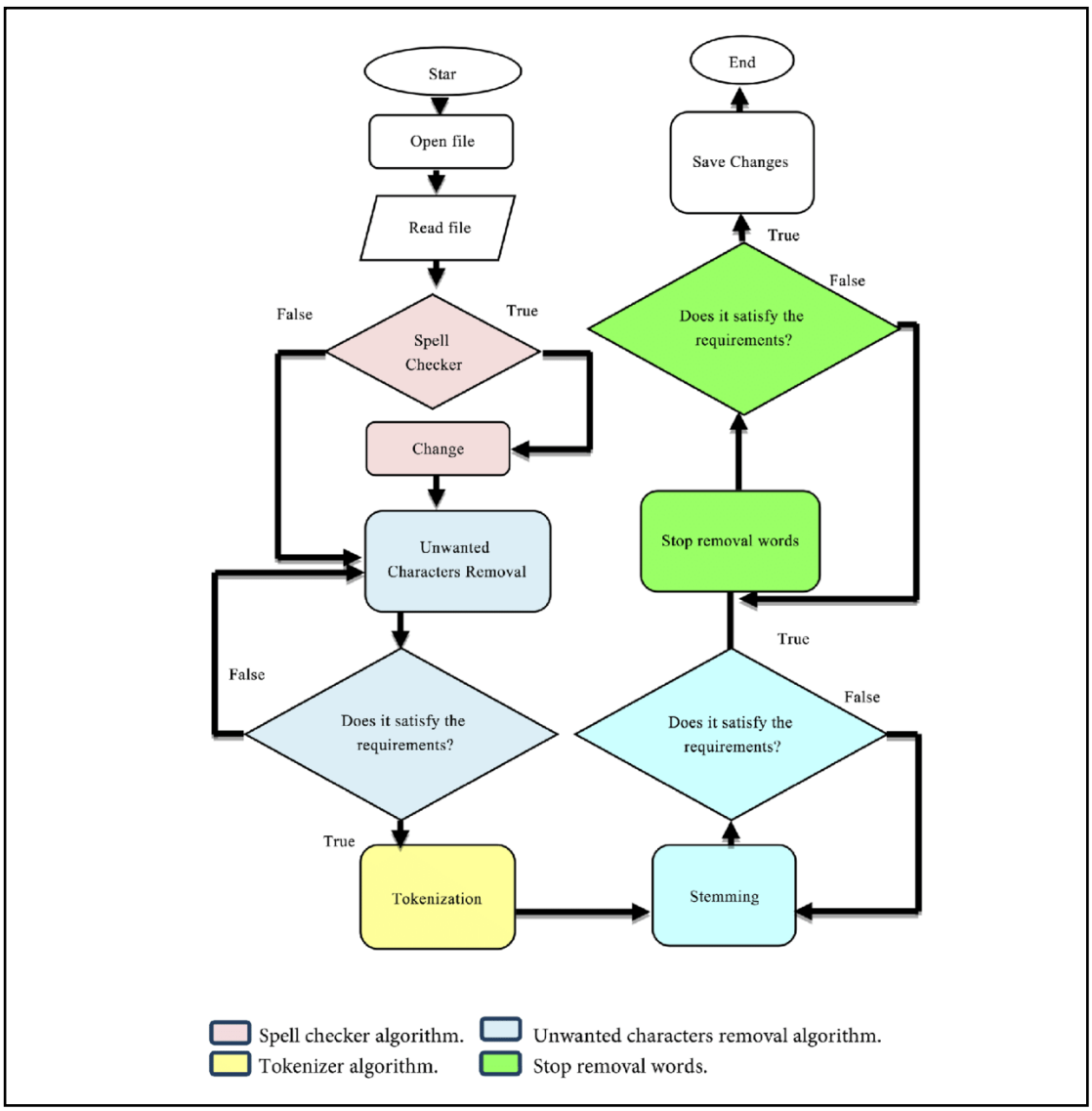

Fig 4.3: Spell Check Flow Diagram

- **Text Translation**

This feature provides translation of English sentences to Spanish, Hindi, German and French. This can be achieved by importing translate in Python. Translate is a simple yet powerful tool written in python with support for multiple translation providers. It can be used as a python module or as a command-line tool. In this system, the user is asked to enter the text or a sentence in the required language text area. This text and the required language are then translated by the translation module and the output is provided in the language desired by the user.

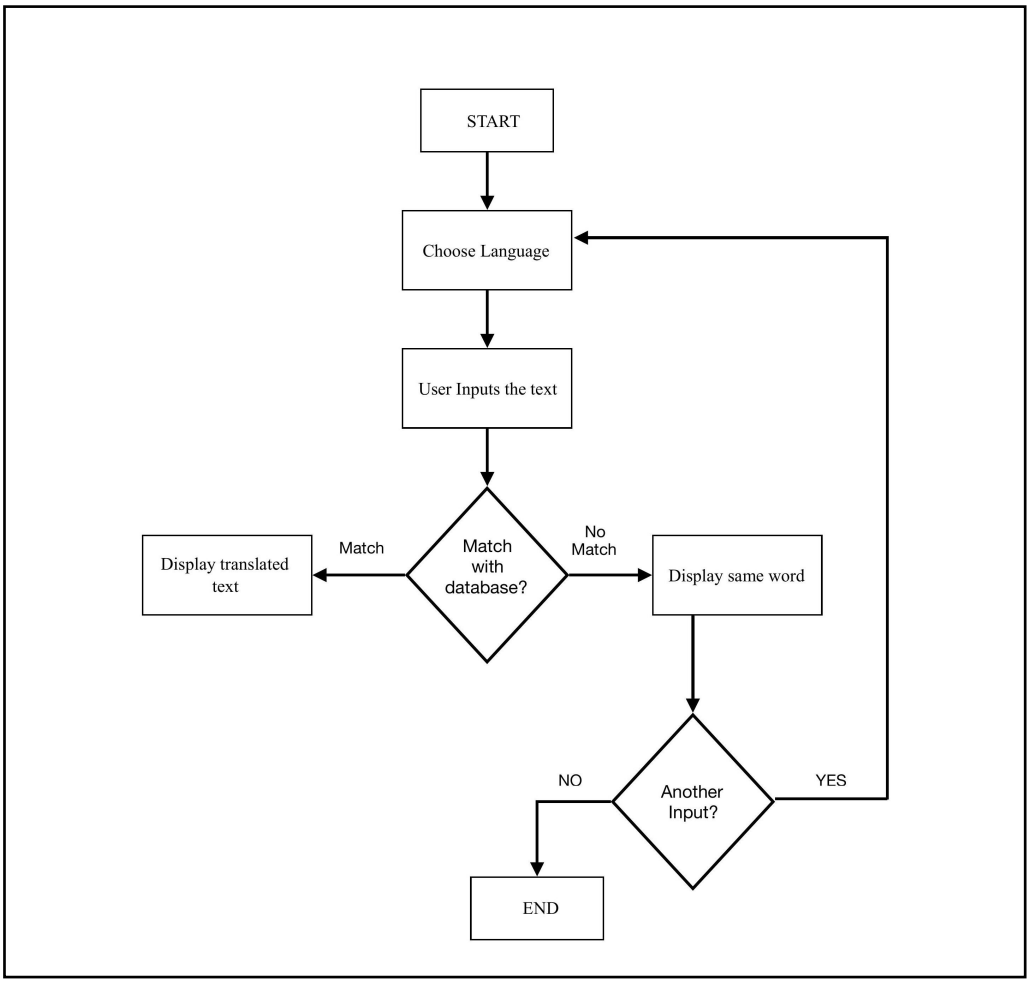

Fig 4.4: Translation Flow Diagram

- **Integration Of Python Script with HTML using Django**

Django is a Python web framework that encourages clean designs and rapid development. It is fast, scalable and secure. It follows a model-template- view architectural pattern. In the views.py file, the input to the file is accepted through the POST request, later according to the functionalities required the request is rendered. The button present in the HTML file is linked with the python script for that feature which should be executed on button-click in order to give the user the desired output. The url.py file is where the URL for the python script is configured in order to access them on the webpage. From manage.py, a Django server is created, and the webpage is displayed according to the HTML file.

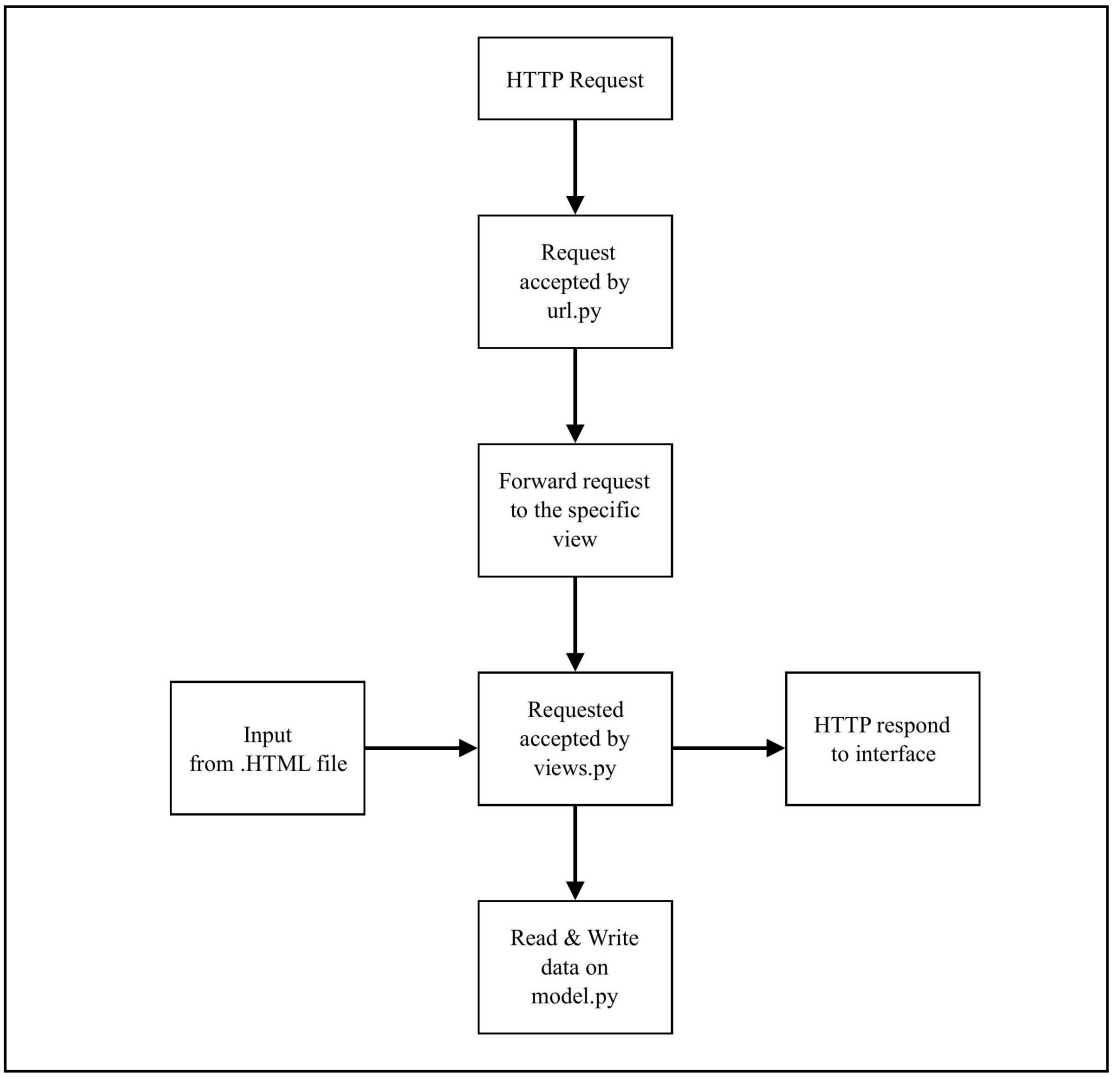

Fig 4.4: Translation Flow Diagram

- **Web-page For User Interface**

The Bootstrap framework proves to be efficient in developing a user interface in the form of a web-page where all these functionalities are available. Each functionality can be used by just clicking on a button. Once the user clicks the button, with the help of Django framework runs in the system and the output is provided in a text box where the user can download it. Django is a python framework for the pragmatic development of web-pages. It utilizes the concept of inheritance from object-oriented programming. By writing limited lines of code one can easily integrate Python with web-pages. The web-page provides each module a different section and can be easily navigated by the user.

4.2 System Code

- **Executable Code for Extraction Of Text From An Image**

```
from PIL import Image
import pytesseract
import argparse
import cv2
import stat
import os

pytesseract.pytesseract.tesseract_cmd = 'C:\\Program Files\\Tesseract-OCR\\tesseract.exe'

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=True,
   help="path to input image to be OCR'd")
ap.add_argument("-p", "--preprocess", type=str, default="thresh",
   help="type of preprocessing to be done")
args = vars(ap.parse_args())

# load the example image and convert it to grayscale
image = cv2.imread(args["image"])
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# check to see if we should apply thresholding to preprocess the
# image
if args["preprocess"] == "thresh":
   gray = cv2.threshold(gray, 0, 255,
      cv2.THRESH_BINARY | cv2.THRESH_OTSU)[1]

# make a check to see if median blurring should be done to remove
# noise
elif args["preprocess"] == "blur":
 gray = cv2.medianBlur(gray, 3)
```

```
# write the grayscale image to disk as a temporary file so we can
# apply OCR to it
filename = "{}.png".format(os.getpid())
cv2.imwrite(filename, gray)

# load the image as a PIL/Pillow image, apply OCR, and then delete
# the temporary file
text = pytesseract.image_to_string(Image.open(filename))
os.remove(filename)
print(text)



# text_file = open('output.txt', 'w+')
# text_file.write(text)
# text_file.close()
#
# print("Your file is ready please check in the FYP folder of desktop file name is output.txt")



# show the output images
cv2.imshow("Image", image)
cv2.imshow("Output", gray)
cv2.waitKey(0)
```

- **Executable Code for Text Summarization**

```
"""Data collection from web through Web-scraping
 Data Cleanup (Like special characters, numeric values, stopwords, punctuations etc)
Tokenization - Creation of tokens (Word Tokens & Sentence tokens)
Calculate Word Frequency for each word by excluding stop words
Calculate Weighted Frequency for each word
Calculate Sentence scores based on each word within sentence
Creation of summary with top 10 highest scored sentences"""

#To scrap the data from the data given like the url or the file.
from urllib import request
from bs4 import BeautifulSoup as bs
import re
import nltk
import heapq
what=int(input("what you want to summarize?\n 1.WebPage \n 2.Text File\n "))

if(what==2):
    filename=input("enter the filename you want to summarize\n")
    f=open(filename,encoding='utf-8',mode="r+")
    paragraphContents = f.read()
    f.close()
    allParagraphContent = ""
    for paragraphContent in paragraphContents:
        allParagraphContent += paragraphContent
else:
    url=input("enter the webpage url you want to summarize\n")
    #url = "https://en.wikipedia.org/wiki/Machine_learning"
    allParagraphContent = ""
    htmlDoc = request.urlopen(url)
    soupObject = bs(htmlDoc, 'html.parser')
    paragraphContents = soupObject.findAll('p')
    for paragraphContent in paragraphContents:
        allParagraphContent += paragraphContent.text
```

```
#print(paragraphContents)
#print(allParagraphContent)

"""here re.sub stands for remove substring, then [[0-9]] stands for removing all square brackets and numbers within it
' '--> this stands for removing all spaces and s+ stands for removing the whitespaces
"""
allParagraphContent_cleanerData = re.sub(r'\[[0-9]*\]',' ',allParagraphContent)
allParagraphContent_cleanedData = re.sub(r'\s+',' ',allParagraphContent_cleanerData)

sentences_tokens = nltk.sent_tokenize(allParagraphContent_cleanedData)
#print(allParagraphContent_cleanedData)
allParagraphContent_cleanedData)
allParagraphContent_cleanedData = re.sub(r'\s+',' ',allParagraphContent_cleanedData)

#creating sentence Tokens

words_tokens     = nltk.word_tokenize(allParagraphContent_cleanedData)

#calculate the Frequency and remove stopwords

stopwords = nltk.corpus.stopwords.words('english')

word_frequencies = {}

for word in words_tokens:
   if word not in stopwords:
      if word not in word_frequencies.keys():
         word_frequencies[word]=1
      else:
         word_frequencies[word]+=1
```

```
#print(word_frequencies)

#calculate weighted Frequency
maximum_frequency_word = max(word_frequencies.values())

for word in word_frequencies.keys():
   word_frequencies[word] =  (word_frequencies[word]/maximum_frequency_word)

#print(word_frequencies)

#calculate sentence score with each word weighted Frequency
sentences_scores = {}

for sentence in sentences_tokens:
   for word in nltk.word_tokenize(sentence.lower()):
      if word in word_frequencies.keys():
         if(len(sentence.split( ))) < 30:
            if sentence not in sentences_scores.keys():
               sentences_scores[sentence] = word_frequencies[word]
            else:
               sentences_scores[sentence] += word_frequencies[word]

#print(sentences_scores)

n = int(input("How many line summary do you want?"))

summary_MachineLearning = heapq.nlargest(n, sentences_scores, key=
sentences_scores.get)
print(summary_MachineLearning)
```

- **Executable Code for Grammar Check**

```
from spellchecker import SpellChecker

spell = SpellChecker()

# find those words that may be misspelled
a=list(input("Enter a sentence  :  ").split())
# print(a)
misspelled = spell.unknown(a)

for word in misspelled:
    # Get the one `most likely` answer
    print(spell.correction(word))

    # Get a list of `likely` options
    print(spell.candidates(word))
```

- **Executable Code for Text Translation**

```
from translate import Translator

print("1. French\n2. Hindi\n3. Spanish\n4.German")
choice= int(input("Enter Choice: "))
text =str(input("Enter text: "))


if (choice==1):
    translator= Translator(to_lang="French")
    translation = translator.translate(text)
    print (translation)

elif (choice==2):
    translator= Translator(to_lang="hindi")
    translation = translator.translate(text)
    print (translation)

elif (choice==3):
```

```
elif (choice==4):
    translator= Translator(to_lang="german")
    translation = translator.translate(text)
    print (translation)

elif(choice==5):
    translator= Translator(to_lang="English")
    translation = translator.translate(text)
    print (translation)
```

CHAPTER 5

RESULTS AND DISCUSSIONS

RESULTS AND DISCUSSIONS

A selection box is provided for every module where the user can choose the functionality he wants to use. Once the user chooses the module, they will be directed to the respective functionality and will be able to enter the image in case of Extraction of Text From an Image, URL or text in case of summarization. Similarly user can enter the text in the text box provided for Grammar check and finally give the text as input and choose the language the user wants as output for text input provided to the system. All the modules are given the feature of saving/ downloading the output respectively.

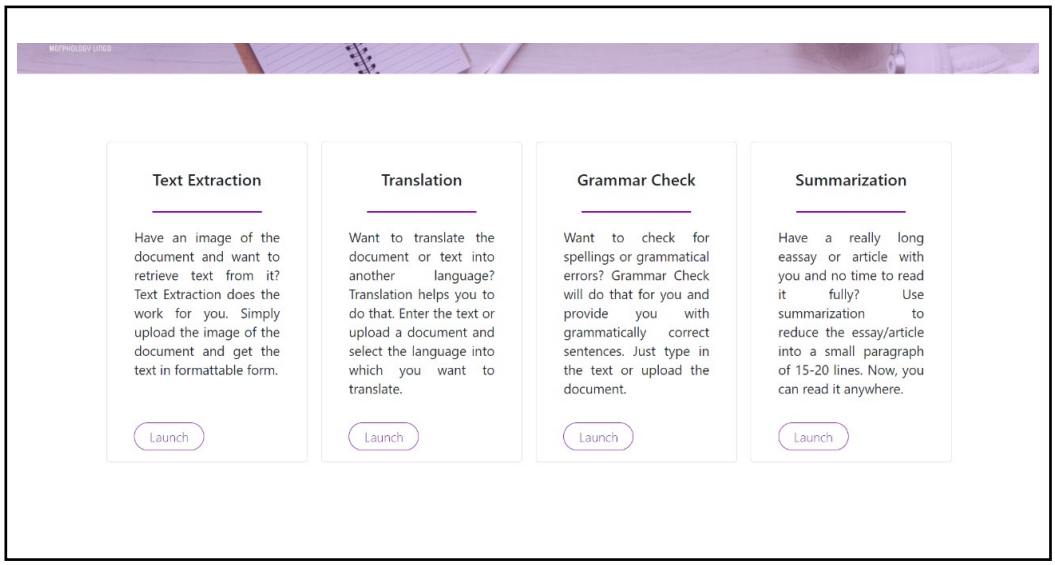

Fig 5.1: Webpage Home

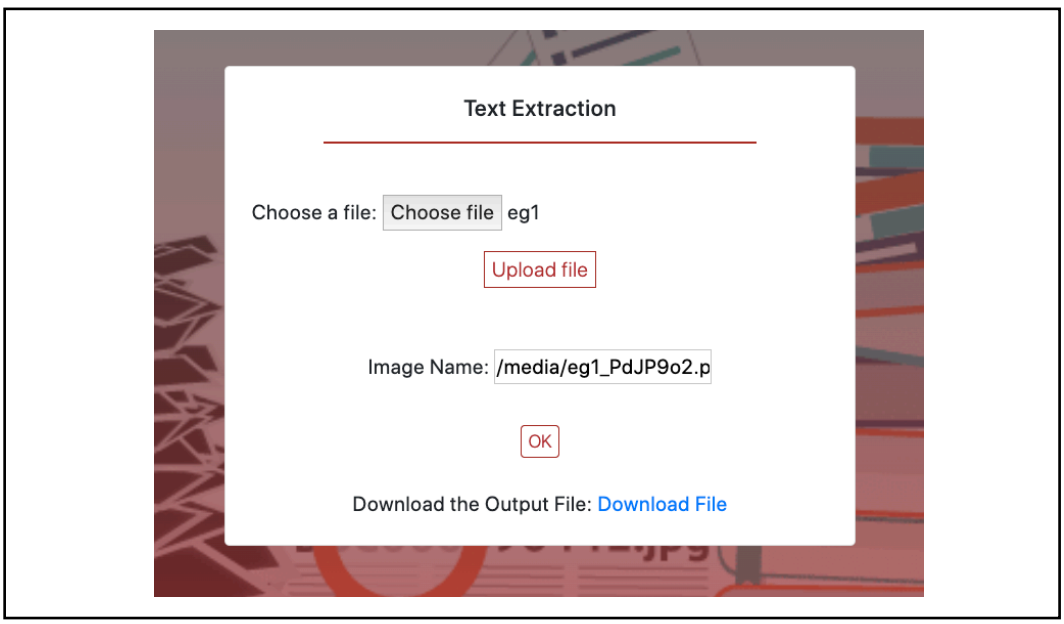

Fig 5.2: Text Extraction Input

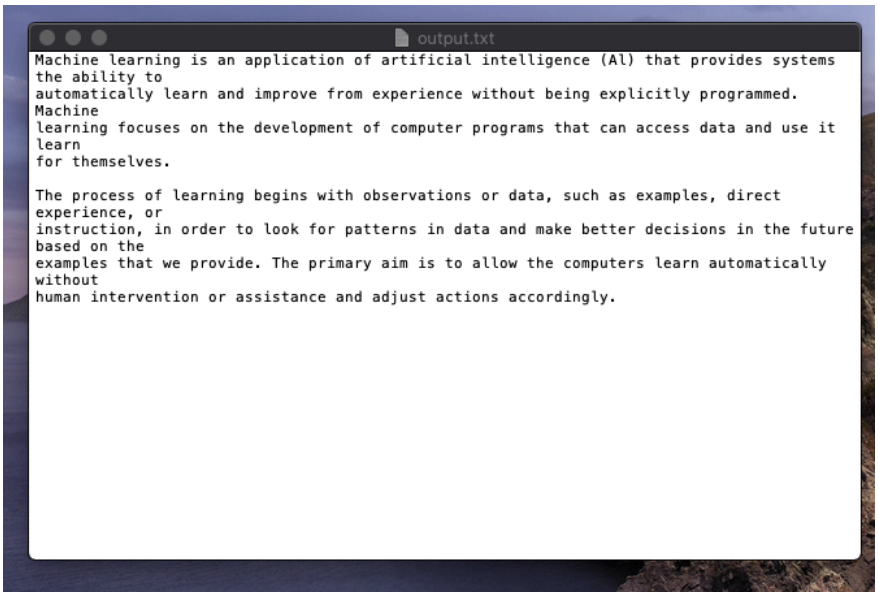

Fig 5.3: Text Extraction Output

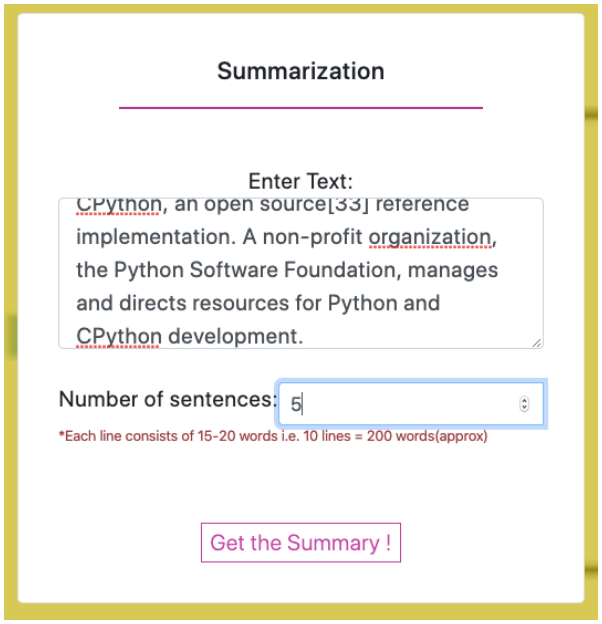

Fig 5.4: Text Summarization of an Article Input

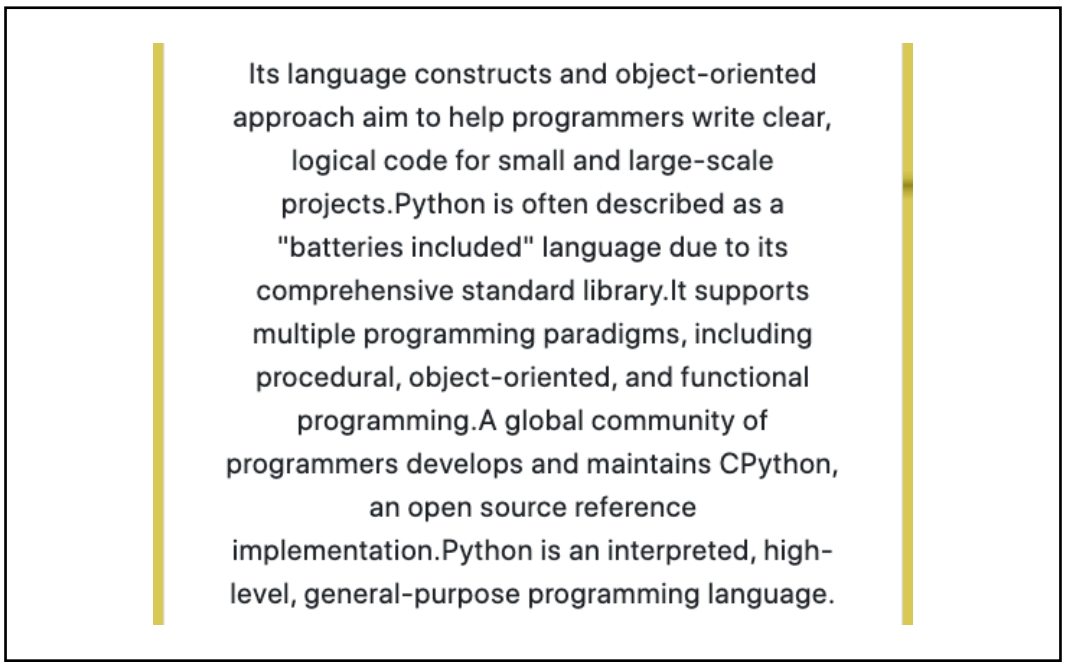

Fig 5.5: Text Summarization of an Article Output

Fig 5.6: Text Summarization of an URL Input

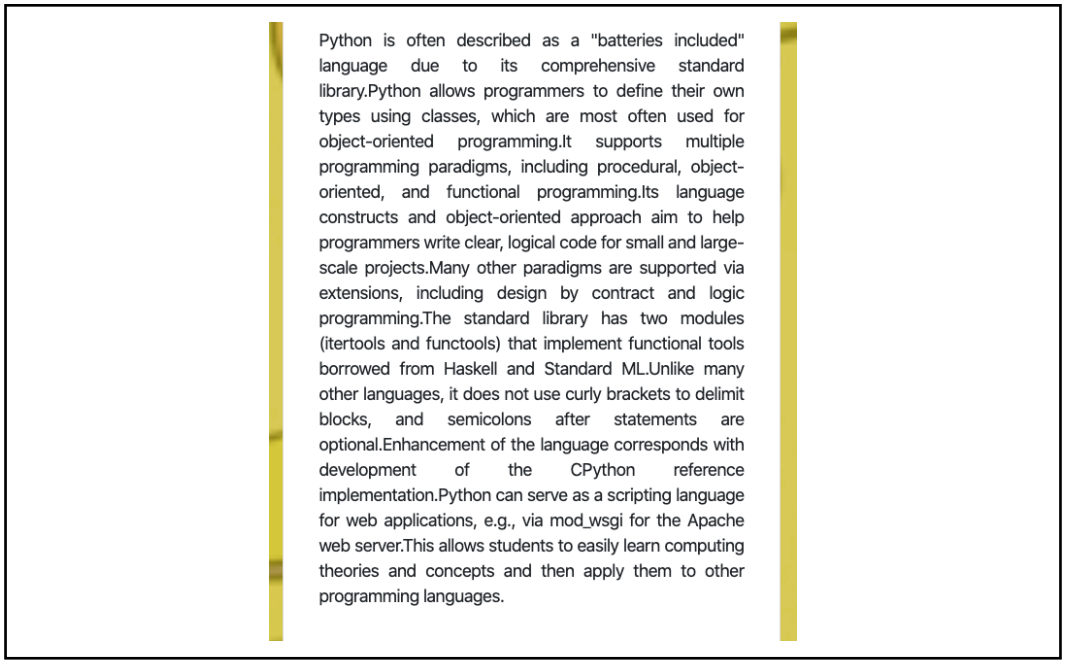

Fig 5.7: Text Summarization of an URL Output

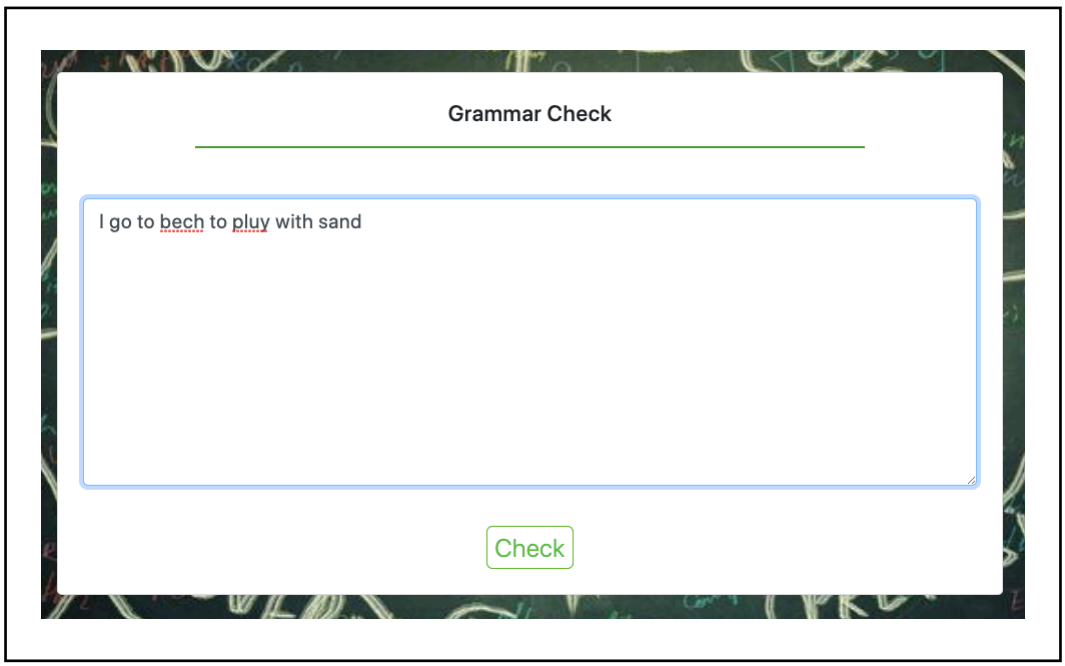

Fig 5.8: Grammar Check Input

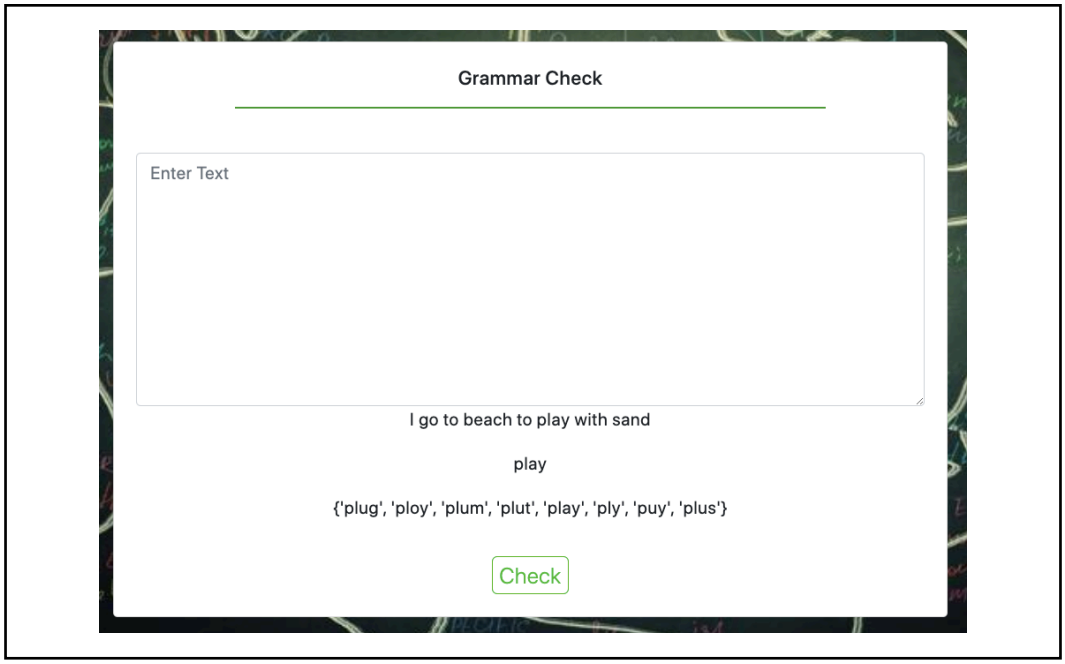

Fig 5.9: Grammar Check Output

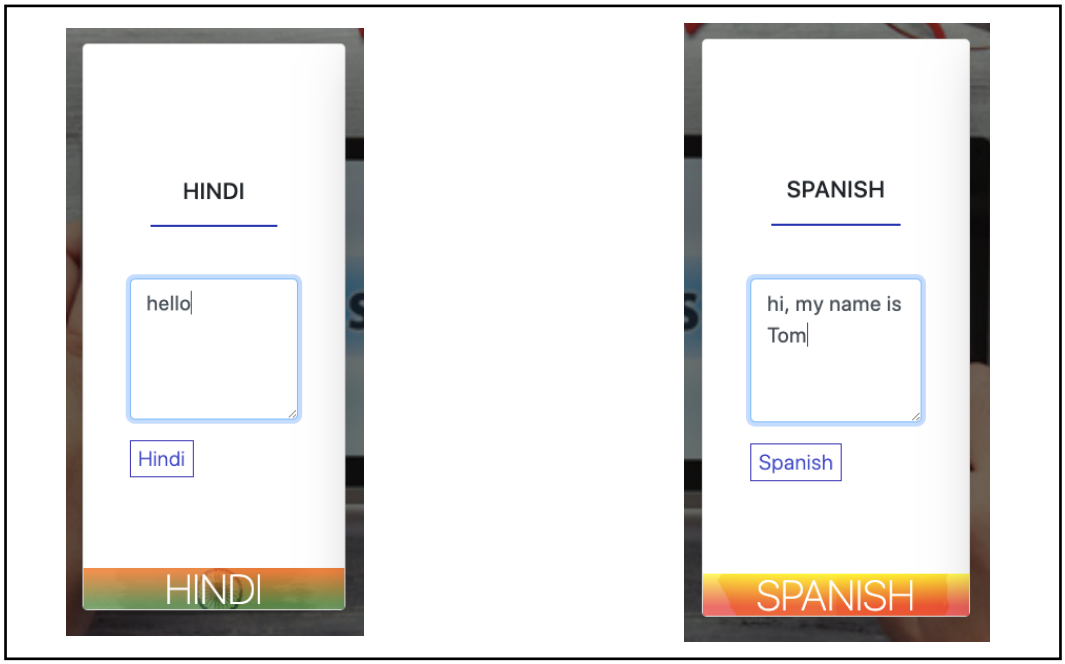

Fig 5.10: Text Translation Input For Hindi & Spanish

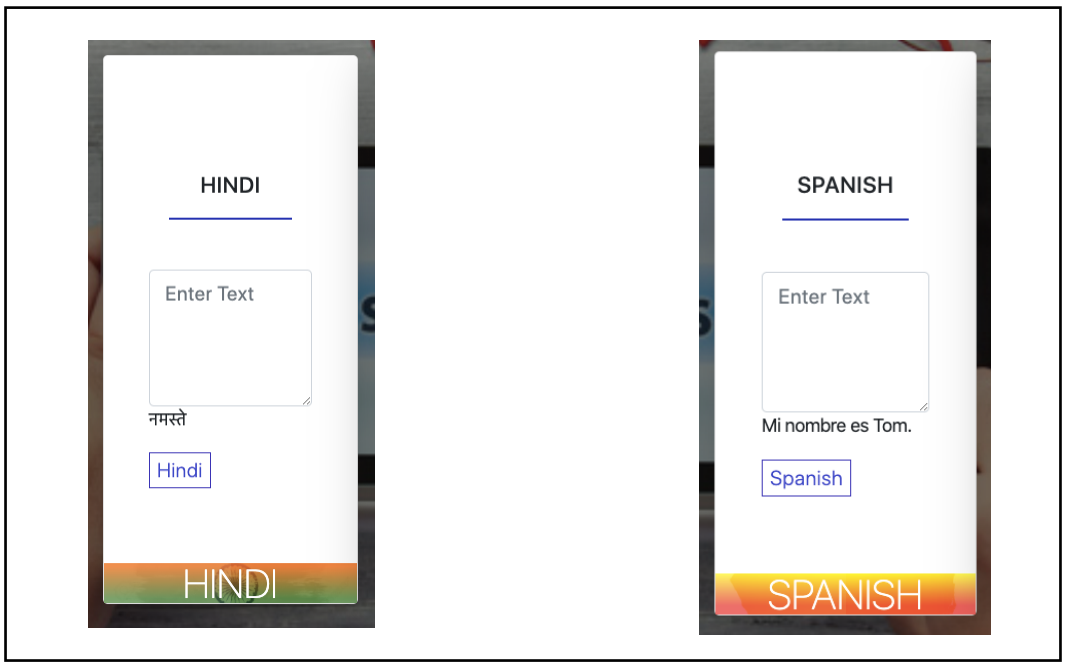

Fig 5.11: Text Translation Output For Hindi & Spanish

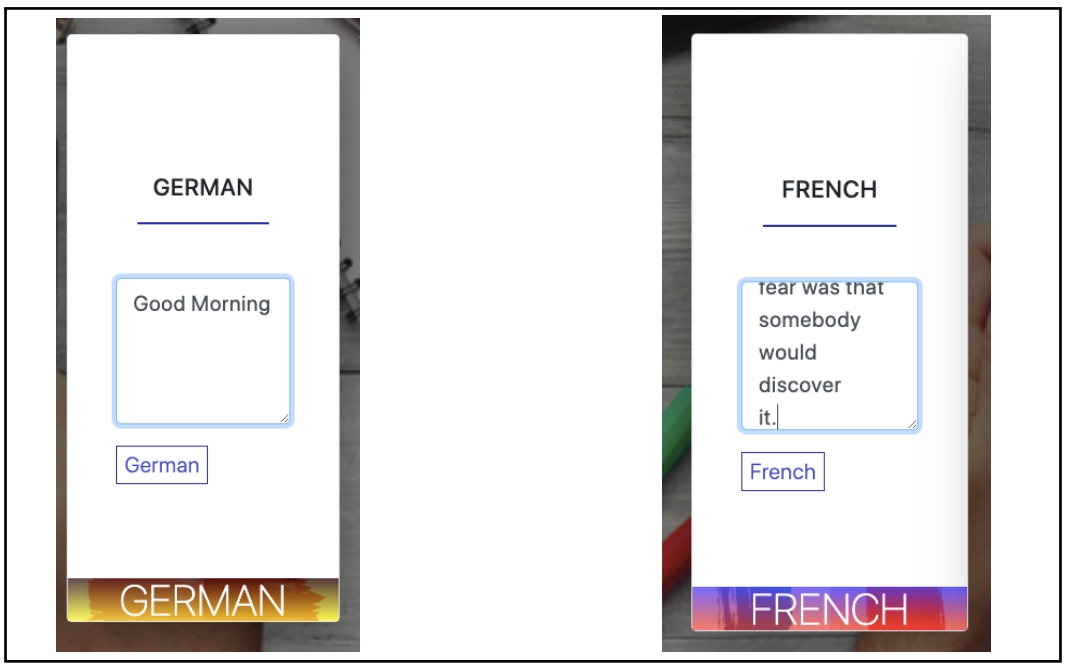

Fig 5.12: Text Translation Input For German & French

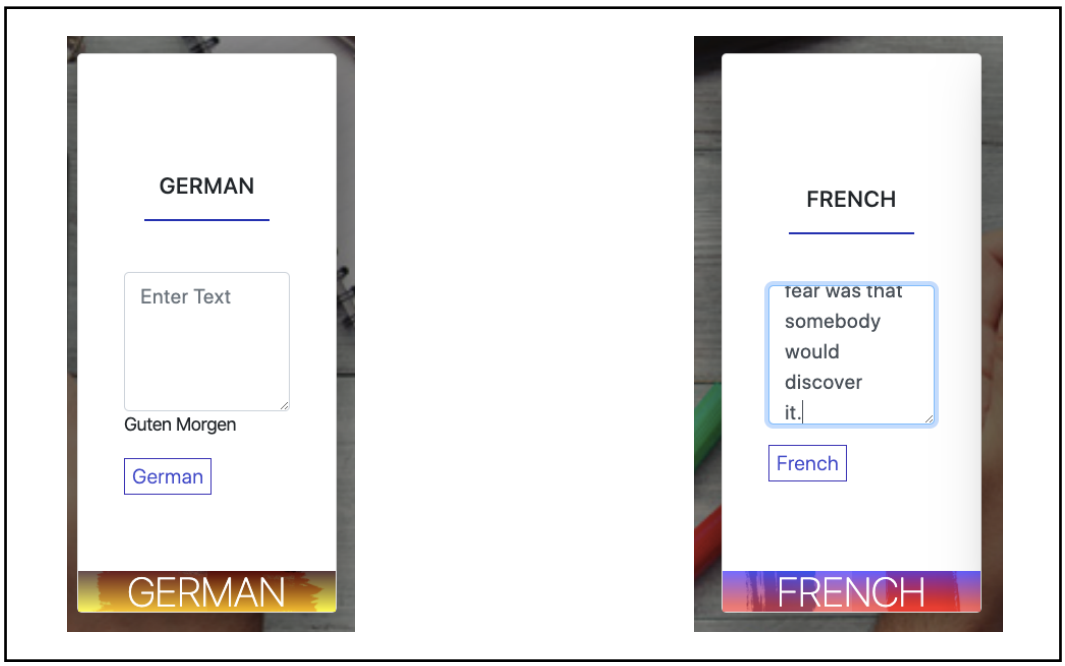

Fig 5.13: Text Translation Output For German & French

CHAPTER 6

CONCERNS

CONCERNS

Q.1 Why is there a need to integrate the modules?

- Most of the people use the given modules one after the other.
- For eg, if a person is visiting a foreign country and wishes to read the sign board, it is not possible because of the language barrier.
- During such instances, the user can extract the text from the image captured of the sign board and translate it on the spot without having to switch applications.
- Hence, integrating the modules proves to be efficient.

Q.2 How does it help?

The system helps in the following ways:

- Saves time
- Effective and Quick
- Can be used in tandem
- It is useful for people from any sector and walk of life

Q.3: How will the user use these modules?

The user is provided a user friendly webpage where he/she can easily choose and go about their task, respectively.

Q.4: How can one use the output of any specific functionality?

The user is provided a save/download option where the output of the specific functionality can be saved in the user's system. The generated output can be downloaded as a ".txt" file.

Q.5: Can any type of image be given as input for the extraction of text module?

Images with appropriate extensions like .jpg or .img are acceptable by the system. The images can be anything from screenshots to real time captured images.

Q.6: Can you enter the link of a particular website for summarisation?

URL's can be given as input to the system. The user can copy and paste the URL of any website and specify the number of lines for obtaining the summary.

Q.7: Can you enter only a single word instead of a sentence in the Grammar check module?

Yes, the user can enter only a single word and the system will check for the spelling and if incorrect, the system will provide a list of suggested spellings out of which the user can choose an appropriate one respectively.

Q.8: From which language to which language does the text translation work?

Text translations can be performed from English to specified languages such as Spanish, Hindi, French or Hindi. The user is allowed to enter text only in English and choose any given language as per his choice of interest.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

In conclusion, this system can be implemented and will prove to be effective for anybody in any walk of life. All the basic text-based functionalities integrated together will ease the work of the user and will save time and energy. It will not only break the ice between people but also make communication and understanding better. Enriched user experience also proves to be one of the most interesting and dynamic factors of this system.

7.2 Future Scope

Improving the system can be done by adding several other modules that further help the user in multiple ways. One of the functionalities that can be added is voice command to the system. For example, if the user commands the system to download the text through his voice, the system can be modelled to understand human language and perform the task commanded by the user. Further, an addition in the grammar check module can be introduced where the system can change the sentence to active or passive voice as per user requirement. This is most useful for students and teachers because the active-passive voice is the basic grammatical rules that are taught. This system can be further enhanced by providing a text file to the user for any handwritten document. As this system eases the work of the user similar modules can be added by identifying the problem of the user respectively.

REFERENCES

REFERENCES

[1] Satish Kumar, Sunil Kumar and Dr. S. Gopinath, "Text Extraction From Images", *International Journal of Advanced Research in Computer Engineering & Technology* Volume 1, Issue 4, June 2012.

[2] K.N. Natei, J. Viradiya, S. Sasikumar, K.N. Natei, "Extracting Text from Image Document and Displaying Its Related Information", *Journal of Engineering Research and Application,* pp 27-33, Vol. 8, Issue5 (Part -V) May 2018.

[3] Sandeep Saini, Vineet Sahula, "A Survey of Machine Translation Techniques and Systems for Indian Languages", *IEEE International Conference on Computational Intelligence & Communication Technology.*

[4] N.Moratanch ,S.Chitrakala, "A survey on extractive text summarization", *IEEE International Conference on Computer, Communication, and Signal Processing,* 2017.

[5] Vibhakti V. Bhaire, Ashiki A. Jadhav, Pradnya A. Pashte, Mr. Magdum P.G, "Spell Checker" , *International Journal of Scientific and Research Publications,* Volume 5, Issue 4, April 2015.

[6] Prof. B Nithya Ramesh, Aashay R Amballi, Vivekananda Mahanta, "Django- The Python Web Framework", *International Journal of Computer Science and Information Technology Research,* Vol. 6, Issue 2, pp: (59-63), Month: April - June 2018.

[7] Sachin Grover, Kushal Arora, Suman K. Mitra, "Checking Parts Of Speech", *IEEE India Council Conference, INDICON*, 20 December 2009.

[8] L. Neto, A. A. Freitas and C. A. Kaestner, "Automatic Text Summarization", *Springer*, pp. 205-215, 2002

PAPER PUBLISHED

Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 01, January-2020

Natural Language Toolkit based Morphology Linguistics

Alifya Khan
Information Technology
Vidyalankar Institute of Technology
Mumbai, India

Pratyusha Trivedi
Information Technology
Vidyalankar Institute of Technology,
Mumbai, India

Karthik Ashok
Information Technology,
Vidyalankar Institute of Technology,
Mumbai, India

Prof. Kanchan Dhuri
Information Technology,
Vidyalankar Institute of Technology,
Mumbai, India

Abstract— In the current scenario, there are different apps, websites, etc. to carry out different functionalities with respect to text such as grammar correction, translation of text, extraction of text from image or videos, etc. There is no app or a website where a user can get all these functions/features at one place and hence the user is forced to install different apps or visit different websites to carry out those functions. The proposed system identifies this problem and tries to overcome it by providing various text-based features at one place, so that the user will not have to hop from app to app or website to website to carry out various functions. The proposed system will provide users with various functions such as grammar checking, text extraction from an image, text summarization, translation of text into different languages, etc. which will help them in their daily life.

I. INTRODUCTION

In this rapidly growing generation and increase in technology as well as productivity of man, now-a-days people want their work to be completed as soon as possible. Therefore, designing a system which will help many walks of life from the corporate sector to students and senior citizens who interact with such technologies on a regular basis.

Most commonly while working on any type of document the writer faces problems like incorrect grammar usage which leads to poor presentation of the document or a high count of lines which tends to miss the core point of the document. Sometimes, a document in a specific language needs to be translated to another language for better user interaction and understanding, this may make the user confused and unable to read the document. Due to this people tend to postpone their work leading to inefficiency.

Now, to get over this one has to use various applications or websites which can be time consuming. This system helps one to perform all the specified tasks at a single website and obtain finalized documents according to one's needs.

II. PROBLEM STATEMENT

In the current situation, there are many websites and applications available which provide different text-based functionalities like extracting text from an image, translation of text to different languages, grammar check, etc. Generally, all these functionalities are used in tandem with each other. For e.g., to read a sign board in a different language, the first step is to extract text from that image and translate it to any respective language as required. Hence, to do this one has to switch from application to application which can be time consuming. To overcome this problem, an integrated environment is built where all these functionalities are available.

III. PROPOSED SYSTEM

The proposed system provides all the text related features at one place to ease the task of the user. With the help of this system the user will easily be able to enhance their sentences, extract text from images, summaries a whole essay or an article and even will be able to translate the sentences into different languages as per requirement. This will not only save the user's time but also help them to provide an efficient output in their workspace.

The proposed system is divided into four main modules namely:

Text Extraction from an Image

This module focuses on extracting text from any kind of image and stores it in a document for usage.

Text Translation

This module is based on translation of text from a given language to any specified language as per user requirement.

Grammar Check

This module will help in correcting grammatical errors in English. Once a user inputs a sentence, grammatical errors are checked, and the corrected sentence is provided as output for further use.

Summarization of Article

This module is based on summarization of an article to specified number of sentences as mentioned by the user

IV. METHODOLOGY

As this system is totally based on text-based functionalities, each module represents different text functionalities which

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 01, January-2020

the user can use. Below different modules, each of variable functions are briefed as follows.

Text Extraction from An Image
This module focuses on extracting text from any kind of image. The extracted image is first converted to the grey scale value and pre-processing techniques are applied. This helps in identifying darker backgrounds which is then blurred based on the level of darkness. Blurring helps in identifying the text in an assorted background and makes the text clearer to perform extraction. Once extraction is performed the system is ready for output. As the output the system will display the text which has been extracted and will be provided in a convertible format of document.

Text Translation
This module is based on translation of text from English to Hindi, French, Spanish and German as per user requirement. The user will be prompted to enter the text in English. This text will be recognized by the system to verify if the text entered is in English. Further, the list of languages as specified will be chosen by the user and conversion of the text is performed by the system. The output will be displayed in the specified language by the user.

Grammar Check
This module will help in correcting grammatical errors in English. Once the user inputs a sentence, the sentence is tokenized which helps in identifying the part of speech. Further based on a generalized pattern the system identifies whether the sentence is grammatically correct.

Summarization of Article
This module is based on summarization of an article. An article generally consists of many lines and precise information are found in the beginning, middle and at the very end of the article. In order to obtain only significant lines is the motive of this module. The user first enters the article for summarization. Upon this the user enters the number of lines to summarize the article. The system then chooses the sentences based on the count of keywords and higher the frequency, higher is the chance of that sentence to be selected. The top keywords are chosen, and logical sentences are made based on the number of lines that the user has specified.

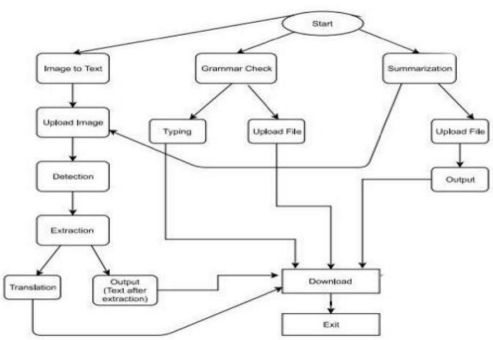

Fig 1. UML Diagram

V. DESIGN TOOLS

Optical Character Recognition:
Optical character recognition or optical character reader (OCR) is used to electronically or mechanically convert the images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo or the text on signs and hoardings.
OCR is used as a "hidden" technology. OCR technology includes data entry automation, indexing documents for search engines, automatic number plate recognition, as well as assisting blind and visually impaired persons.
OCR technology has proven immensely useful in digitizing historic newspapers and texts that have now been converted into fully searchable formats and had made accessing those earlier texts easier and faster.

Open Source Computer Vision:
Open Source Computer Vision (OpenCV) is an open source computer vision and machine learning software library. It was designed to provide the same infrastructure for all computer vision applications. The library has optimized algorithms. These algorithms are used to detect and recognize faces, identify objects, classify human actions in videos, track moving objects, extract 3D models of objects, stitch images together to produce a high resolution image, remove red eyes from images taken using flash, follow eye movements, recognize scenery, etc.

Natural Language Toolkit:
Natural Language Toolkit (NLTK), is a set of libraries and programs for natural language processing (NLP) for English written in the Python programming language. NLTK includes graphical demonstrations, sample data and underlying concepts behind the language processing tasks supported by the toolkit. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet along with text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities.

VI. FEASIBILITY STUDY

Technical Feasibility
Technical feasibility focuses on the technical resources (software and hardware) available for the project. It also helps to determine whether the team is capable of converting the ideas into working systems. The software required for the system are a basic text editor like Atom, Sublime Text for writing the code and using Python as the programming language. The hardware components are not required for this system.

Economic Feasibility
As this system is totally based on software does not require any hardware, the budget of the project is null. Hence, it is highly cost effective.

Operational Feasibility
This assessment involves a study to analyze and determine how well the organization's needs can be met by completing the project. The main objective of the project is to firstly

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 01, January-2020

detect text from an image, translate a given text in English language to any specified language as per the user. Secondly, correct grammatical errors of any text in English language and summarize an article to reduce the number of lines and increase the reader's understanding and speed. As all these technicalities mentioned above are present in one system, it is very useful not only for students but for teachers and working professionals as well.

VII. CONCLUSION

In concluding this entire system, one can be assured that their basic text-based difficulties can be solved by this system. One is just a click away to overcome their problems and present a complete document of any size and top quality which will be legible and easy to understand. Most of our day to day text-based difficulties are overcome by this system. Most useful for students and teachers, this system will also prove to be helpful for corporate people and common people in using modules like extracting text from an image or translation. The user will operate in a user-friendly environment by just having a basic knowledge of computer technology.

VIII. FUTURE SCOPE

As all the modules and different applications provided by this system are text based, various text-based functionalities can be deployed together, one of them being handwriting recognition. In this functionality the system will be able to identify handwritten text and provide a text document of the same. The system will scan the handwritten document and identify each letter according to the English alphabets and provide the output as text document respectively. Further, this system can be integrated with various other features like for instance reading a document to the user. Here, the input document will be provided to the system where the system will recognize each word and prompt it to the user. Another feature that can be added is commanding the system to make changes in the document through voice instruction. For example, prompting "Copy this document", the system will be able to recognize the voice and follow the command and perform it successfully. Finally, converting sentences to active and passive voice can also be added as an extra feature. This functionality will mostly be useful for students and teachers in the literary background as active and passive voice are basic grammatical rules in English language. Thus, including the modules will not only enhance the system but also provide useful and time saving functionalities.

REFERENCES

[1] http://cs229.stanford.edu/proj2018/
[2] https://towardsdatascience.com/build-a-handwritten-text-recognition-system-using-tensorflow-2326a3487cd5
[3] https://pyimagesearch.com/2018/08/20/opencv-text-detection-east-text-detector/
[4] https://github.com/ayesha92ahmad/NLP-image-to-tex
[5] https://github.com/Prashant047/text-extract-from-image
[6] https://www.youtube.com/results?search_query=text+extraction+from+image+using+python
[7] https://www.pyimagesearch.com/2018/09/17/opencv-ocr-and-text-recognition-with-tesseract/
[8] https://www.pyimagesearch.com/category/optical-character-recognition-ocr/
[9] https://www.nltk.org/book/ch08.html
[10] https://www.pyimagesearch.com/2017/07/10/using-tesseract-ocr-python/
[11] https://en.wikipedia.org/wiki/Sentence_diagram
[12] https://www.youtube.com/watch?v=nRBnh4qbPHI&feature=youtu.be
[13] https://realpython.com/natural-language-processing-spacy-python/
[14] http://www.danielnaber.de/languagetool/download/style_and_grammar_checker.pdf
[15] https://www.quora.com/What-kind-of-algorithms-is-Grammarly-using-to-grammar-check/answer/Khushal-Singh-1?ch=3&share=fad573d0&srid=pSniA
[16] https://www.youtube.com/watch?v=Dcvecpq7N0I
[17] https://www.kdnuggets.com/2017/09/machine-learning-translation-google-translate-algorithm.html

CERTIFICATIONS

IJERT International Journal of Engineering Research & Technology
ISSN : 2278 - 0181, www.ijert.org
(Published by : ESRSA Publication)

Certificate Of Publication

This is to certify that

Pratyusha Trivedi

Has published a research paper entitled

Natural Language Toolkit based Morphology Linguistics

In IJERT, Volume 9, Issue 1 , January - 2020

Registration No: IJERTV9IS010320 **Date:** 07-02-2020 **Chief Editor,IJERT**

International Journal of Engineering Research & Technology

IJERT International Journal of Engineering Research & Technology
ISSN : 2278 - 0181, www.ijert.org
(Published by : ESRSA Publication)

Certificate Of Publication

This is to certify that

Alifya Khan

Has published a research paper entitled

Natural Language Toolkit based Morphology Linguistics

In IJERT, Volume 9, Issue 1 , January - 2020

Registration No: IJERTV9IS010320 **Date:** 07-02-2020 **Chief Editor,IJERT**

International Journal of Engineering Research & Technology

IJERT International Journal of Engineering Research & Technology
ISSN : 2278 - 0181, www.ijert.org
(Published by : ESRSA Publication)

Certificate Of Publication

This is to certify that

Karthik Ashok

Has published a research paper entitled

Natural Language Toolkit based Morphology Linguistics

In IJERT, Volume 9, Issue 1 , January - 2020

Registration No: IJERTV9IS010320 **Date:** 07-02-2020 **Chief Editor,IJERT**

International Journal of Engineering Research & Technology