Image Processing and NLTK based Text-functionalities on a Single Platform

Pratyusha Trivedi, *Alifya* Khan, *Karthik* Ashok and Prof. *Kanchan* Dhuri.

Vidyalankar Institute of Technology, Department of Information Technology, Wadala, Mumbai, India.

Abstract: In the current scenario, there are various applications and websites that provide extraction of text from an image, translation of the text to different languages, summarization of articles or essays and correcting spellings or grammatical errors in a sentence or document. However, these functionalities are not available on a single platform and hence the user must hop between various sites or is forced to install different applications to use them, which is a hectic task. This system identifies this problem and tries to overcome it by integrating all the above-mentioned functionalities on an independent platform which will save time and effort of the user and ease their task.

1. Introduction

Every individual today leads a fast life. Even most of the tasks are made convenient as they contribute largely to save one's time, money and energy. Right from corporate staff and businessmen to students and senior citizens, everybody is bound to use text-related features. Usually, people make various documents and face the problem of the correct use of grammar. Reading long articles or essays might be stressful and time-consuming also, not being able to understand a document or a text due to the language barrier may lead to delay in work. Similarly, being able to use the text in an image for any of the above purposes is not possible under a single frame. This system identifies the above problems and provides a solution in an all in one webpage. With the help of this system, one can easily navigate through different functionalities as per their usage. The system is largely divided into four modules. The first module deals with the extraction of text from an image that provides the functionality of obtaining the text within any image. The second module helps in the summarization of any article or an essay to the desired number of sentences as per user input. Furthermore, the third module deals with the translation of text in the English language to other languages such as Spanish, German, Hindi, and French as they are the most commonly used languages around the world. Finally, the fourth module helps the user in correcting the grammatical errors and spelling mistakes within a sentence.

2. Problem Statement

Nowadays, there are many websites and applications available that provide different text-based functionalities like extracting the text from an image, translation of text to different languages, grammar check, etc. Generally, all these functions are used alongside each other. For e.g., when a person has a hardcopy of a document and needs to make some changes to it, he/she first needs to click the picture of the document then extract the text from that image. The extracted text is then made available to the user in .pdf. form where it is impossible to make the changes. 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 to delay and inefficiency. Also, reading long lines of articles may not be feasible in many situations due to time constraints. These different modules are the most basic and useful in day to day life of a user and hence it is important to identify them. This system has been developed by considering all the above problems and understanding the user's needs and purpose. Therefore, an integrated environment is built where all these functionalities are available at one place.

3. Literature Survey

Each of these functionalities has already been implemented by various methods. It is very important to implement these modules first in order to integrate them. Implementation of each of these models and integration of the same can be achieved in many ways.

The method used in [1] to extract the text from an image follows five steps. The first step is to convert the coloured image to grayscale image. Here, the coloured images which can follow the RGB or CMYK pattern which include information of each of the coloured pixels is converted to greyscale 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 are checked and the transition of pixels horizontally and search vertically for finding the area of text. Specific heuristics are determined for finding the location of the text. In the final step, the disturbed pixels are rearranged for clarity.

In the case of the translation of the text, the author mentions the different techniques that are available for machine translation (MT) [3]. 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 Hybrid, 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 language can be converted to any language in many ways.

Summarization is the most useful and can be performed in many ways. Broadly, there are two major ways, unsupervised learning and supervised learning [4]. Unsupervised learning has four approaches. The first one focuses on a graph-based approach which is highly useful in document-based summarization. The sentences are assigned as a graph and the edges are assigned cosine values. Based on these values the Lex Rank 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 based on how they 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, it is very important to first correct the spellings of words within a sentence. 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 [5]. Further, by splitting the sentence, identifying the part of speech and comparing it with the general Subject+Verb+Object pattern the words are rearranged and the sentence is corrected.

The Django framework is one of the fastest frameworks nowadays. It is quick, secure, flexible and is a benefit to both developers and users at the same time. 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. [6]

4. Methodology

This system is totally based on providing text-based functionalities on a single platform. Each module represents features such as extraction of text from an image, translation of the text to other languages, summarization and grammar checking and are combined for user’s ease. Below each of these functionalities, the framework and the techniques are briefed as follows:

4.1. 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 grayscale. By using pytesseract.image_to_string we convert the contents of the image into our desired string then we pass a reference to the temporary image 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.

4.2. Summarization

This module focuses on shortening the length of an article while maintaining its gist 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 the sentence. Here, first, the input is taken from the user, the input can be either a plain text entered by the user or a web page text which is being scraped using beautiful soup. Once the input is stored, the input data is cleaned (cleaning is done on special characters, numbers, punctuations, and stop words). Then at the later stage, the tokenization is done i.e. the words and sentences are tokenized. Then word frequency and weighted frequency for each word by excluding stop words is calculated. At last, the sentence scores based on each word within the sentence is calculated. As the sentence length limit is set to 20, no sentence will exceed this length in the summary. The user will be prompted to enter the number of lines and based on the frequency of words used in a sentence, sentence scores are determined and the sentences with higher scores are chosen which provide the accurate summary.

4.3. 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.

4.4. 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.

4.5. Integration of Python scripts 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.

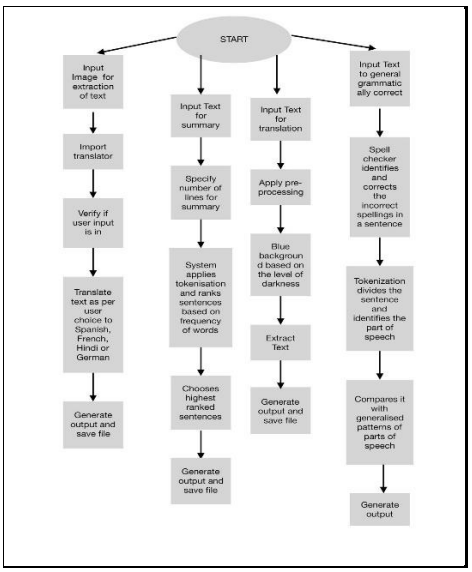

Fig. 4.1 Methodology

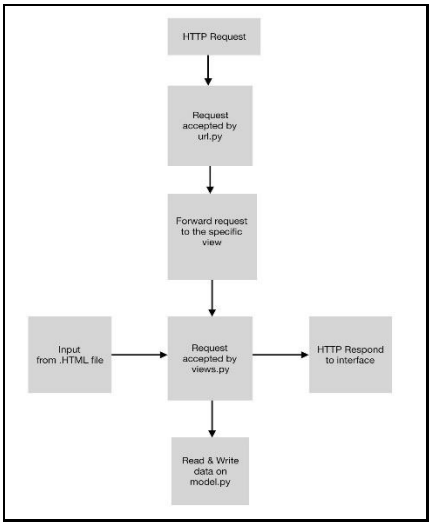

Fig. 4.2. Django Workflow

5. RESULTS

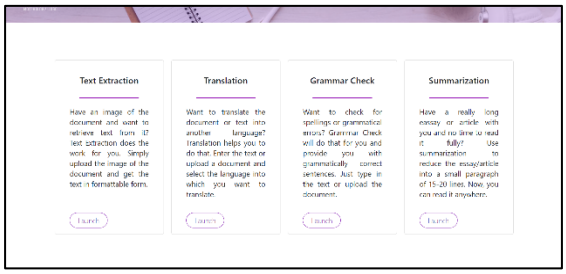

Fig. 5. Home Page.

5.1 In the Text Extraction Module:

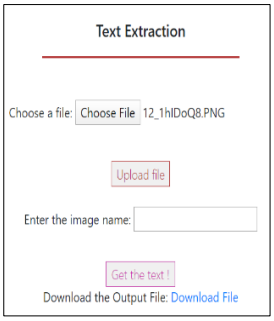

Fig. 5.1.1. Choose a file.

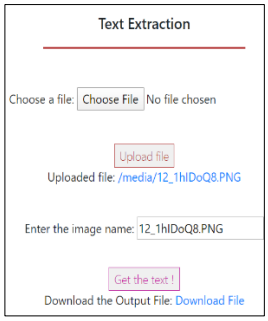

Fig. 5.1.2. Upload a file.

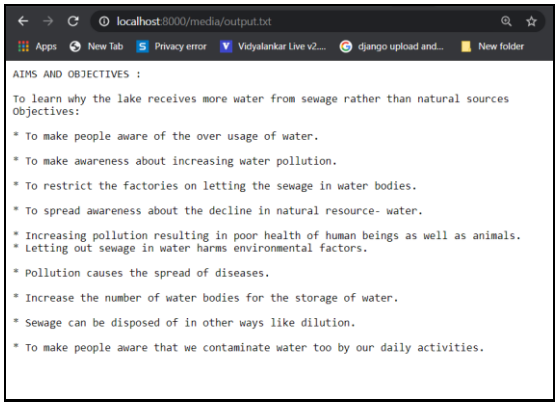

Fig. 5.1.3. The output of Text Extraction.

5.2 In the Translation Module:

Fig. 5.2.1.
Output of Hindi.

Fig. 5.2.2.
Output of German.

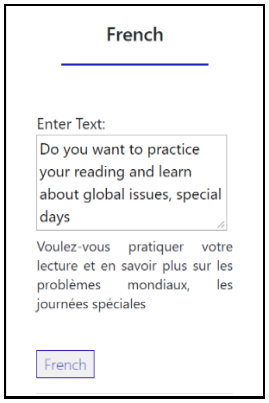

Fig. 5.2.3.
Output of French.

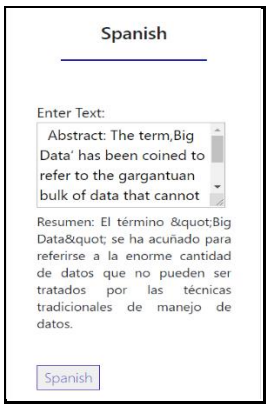

Fig. 5.2.4.
Output of Spanish.

5.3 In the Grammar Check Module:

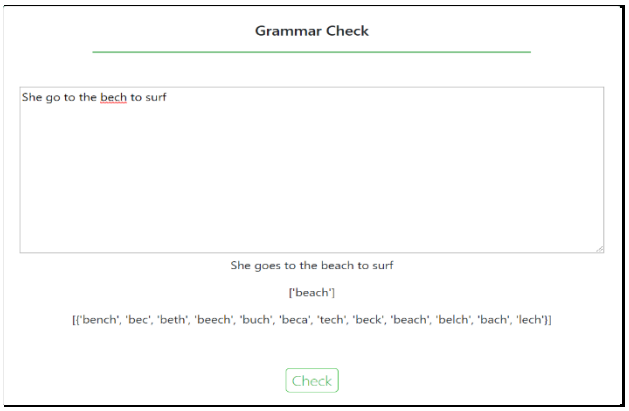

Fig. 5.3.1 Output of Grammar Check.

5.4 In the Summarization Module:

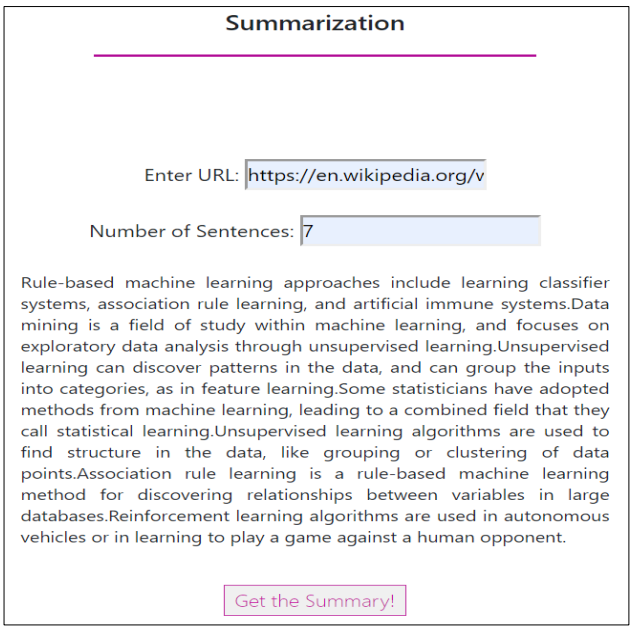

Fig. 5.4.1. The output of Grammar Check.

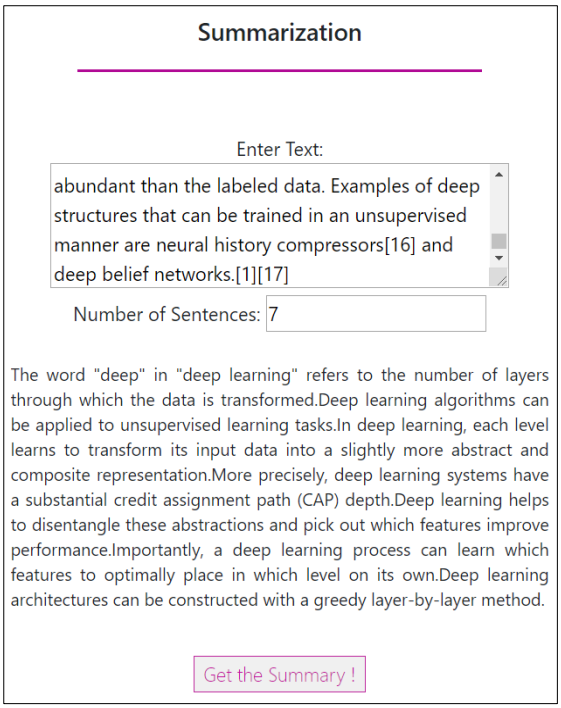

Fig. 5.4.2. Output of Summary of entered text.

6. Conclusion

In conclusion, this system can prove to be effective for anybody in their everyday life. All the basic text-based functionalities are just one click away as the features are integrated together on a single independent platform. It will not only make the work of the user easy but will also save their time and energy. Enriched user experience also proves to be one of the most interesting and dynamic factors of this system. Anybody, ranging from students to teachers to corporate workers, can use this system effectively with minimum or basic knowledge of computer technology.

7. Future Scope

Improving the system can be done by adding several other modules that will further help the user in multiple ways. One of the functionalities that can be added his/her voice command to the system. For example, if the user commands the system to download the text through his/her 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

[1]Satish Kumar, Sunil Kumar and Dr. S. Gopinath, International Journal of Advanced Research in Computer Engineering Technology Volume1, Issue 4, June 2012.

[2] K.N. Natei, J. Viradiya, S. Sasikumar, K.N. Natei Journal of EngineeringResearch and Application, pp 27-33, Vol. 8, Issue5 (Part -V) May 2018. [3] Sandeep Saini, Vineet Sahula, IEEE InternationalConferenceonComputationalIntelligenceCommunicationTechnology,DOI:10.1109/CICT.2015.123, 2015.

[4] N.Moratanch ,S.Chitrakala, IEEE International Conference on Com- puter, Communication, and Signal Processing, 2017.

[5] Vibhakti V. Bhaire, Ashiki A. Jadhav, Pradnya A. Pashte, Mr. Mag- dum P.G, International Journal of Scientific and Research Publications, Volume 5, Issue 4, April 2015.

[6] Prof. B Nithya Ramesh, Aashay R Amballi, Vivekananda Mahanta, 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, IEEE India Council

Conference, INDICON 2009, 20 December 2009.

[8] L. Neto, A. A. Freitas and C. A. Kaestner, Springer, pp. 205-215, 2002. [9] P. S. Giri, International Journal on Advanced Computer Theory and

Engineering, pp.66-71, 2013.

[10] S. K. Dwivedi and P. P. Sukhadeve, Journal of computer science, vol.6, no. 10, p.1111, 2010.

[11] Abhishek Bera, Shubham Darda, Ashish Gaikwad, Shivam Chanage, Prof. Suresh Rathod, International Journal of Advanced Research in Computer Science and Software Engineering Research Paper, Volume6, Issue 5, May 2016.

[12] M. Aniche, G. Bavota, C. Treude, M. A. Gerosa, and A. van Deursen, Empir. Softw. Eng., vol. 23, no. 4, pp. 2121–2157, 2018.