Data Summarization

Satvik Dandale Rakshanda Mahajan Vedanti Vyawahare Prathamesh Dhawale

Computer Department, Vishwakarma Institute of Technology (Affiliated to Savitribai Phule Pune University)
666, Upper Indira Nagar, Bibwewadi, Pune, Maharashtra, India

satvik.dandale17@vit.edu
rakshanda.mahajan17@vit.edu
vedanti.vyawahare17@vit.edu
prathamesh.dhawale17@vit.edu

***Abstract*— There have been many uses of Natural Language Processing in recent days. With the growth in the popularity of Python and its wide range of libraries, NLP has got its way into Python implementations. Many of these implementations focus on the areas like Voice Recognition or Text to Speech. But these applications do not focus on the less obvious but important aspect of our daily lives. Any particular day, we come across huge amount of data to read and process in a short time. This is where this project exploits the need of Content Summarization to reduce the amount of data without actually losing its main objective. This project applies Natural Language Processing to save the time and efforts of the reader.**

***Keywords*— frequency, summarization, search engine, natural language processing.**

I. Introduction

This paper reflects the significance of basic text summarization in our daily life. The texts that are being summarized are part of the huge texts of any topic that we encounter and take quite a lot of time to process. Auto Text Summarisation uses a set of techniques that make it exceptionally good at producing appropriate output. These techniques are based on Natural Language Processing that can be used to use and manipulate text the way we want.

Natural Language Processing is a branch of Computer Science and Artificial Intelligence that is concerned with the interaction of Computers with natural (Human) Language. In particular, how to program computers to process and analyse large amounts of natural language data. This project generally uses the part of NLP called as Part-of-speech tagging, which determines the part of speech for each of the word in the text.

When automated, Text Summarization can be used in various parts of our daily aspects of life. From newspaper articles to story books, we can summarize the texts to save time by reading the short, meaningful summary of the texts instead of reading the whole thing.

This project also demonstrates the importance and the application of NLP in Search Engine Optimization. SEO is used while developing websites to optimise their indexing score decided by a Search Engine so that this website should be indexed at a higher level. Indexing focuses on the frequent keywords used in the website or the various other keywords present in the description or title tags of the website. The application focuses on what the description tag of the website should contain so that maximum indexing score can be achieved.

II. Ease Of Use

A. Saving Valuable Time:

The main objective of this project is to implement Text Summarization in an efficient way so that the user may not have to waste time reading the whole contents, which might be huge in terms of the size.

B. Significant reduction in text size:

This project summarizes the text given to it in a significantly efficient way. The output summary generally ranges from 40 % – 50 % of the original text size, which not only saves time but also reduces the complexity in the text without affecting the meaning of text.

C. Search Engine Optimised Website tags:

As a demonstration of how widely NLP can be used, the current project suggests the description tags to web developers to use in their websites for better indexing by search engines. This will cause the website to get displayed first in the search results when a relevant keyword is searched.

III. Natural Language Processing Approach

Python features a complete NLP library named Natural Language Toolkit. This library package features various operations and classification techniques on text. The NLTK resources used are:

1. nltk.corpus.stopwords: This is an English Dictionary of the words that are most commonly used and are of almost no importance in a sentence while determining the importance of the whole sentence.
2. nltk.tokenize: There are two tokenizing functions used in the projects, sent_tokenize and word_tokenize. These functions segregate the text into individual sentences or words and saves a list of these segregated items.
3. nltk.pos_tag: This is used to determine the part-of-speech of each word, like Noun, Verb, Adjective, etc.

❖ Algorithm:

```
1. Create a table named word frequency, word list and also include stop words. Transverse through the paragraph:
   i.   If current word is not a stop word then increment frequency by 1.
   ii.  Finding the average frequency of words (total frequency/total no of words).
   iii. If frequency used>average then store the word in word list.
2. Create a table for sentence frequency and store each frequency = 0.
3. Transverse through the paragraph:
   i.   If the sentence contains a number then freq= freq+1.
   ii.  If the sentence contains imp word then freq=freq+1 for each word.
   iii. If the sentence contain wordlist then frequency=freq+1 for each word
4. Find total words in paragraph.
```

```
5. Find average length=total no. of words/no. in sentence.
6. For each sentence: Final score = Initial score + length of the sentence
7. Find total frequency=add all final score of each sentence.
8. Find average frequency= total frequency/no. of sentence.
9. If sentence frequency > avg frequency:
        Extract that sentence.
10. Print the extract.
```

IV. Search Engine Optimisation

SEO has become a heavily buzzed topic in the industry of website development. Naturally every website developer would want their site to have more user traffic and more reach. The most common way people access any website is through a search engine where they search the keywords and get most appropriate result that matches with that keyword.

The most common way Search Engines index the website is through the keywords present in the site that occur frequently, various html tags present in the site, etc. The more accurately we design the website, the more rank the page will get and more are the chances of the page to appear in the search results.

When a website is being developed, there are some things that the developer must consider. For example, the 'meta: description' tag of the website must contain those keywords that are very important to the context of the site. This is used by the Search Engine to properly index the website. If the developer makes a description good enough, the rank of the website may naturally increase. This is where NLP comes into play. Using web scraping and NLP processes, we will suggest some description(s) to the developer that he/she might want to use for the 'meta: description' tag.

A. Extracting the web page*:*

The first part of suggesting a better description starts from scarping the web page. The implementation begins by using the python library 'requests' to get the page source code of the given URL of the page.

Then the source code is parsed using 'beautiful soup' python library so that desired information can be extracted.

B. Extracting text*:*

After parsing the code though beautiful soup, the important texts are extracted from the html code. The important tags from which texts are extracted are:

1. <h1> tag
2. <h2> and <h3> tag

C. Getting some more information:

Sometimes, just the text that is being extracted might not be enough to make a useful description. Therefore, the next step is to get some more information about the context, from a third-party source, preferably from Wikipedia.

The URL is processed to get the domain name and then this name is used to search in Wikipedia. From the search result obtained from Wikipedia, only the first paragraph is extracted using the same procedure used for <h1> and <h2> tags.

D. Using NLP:

Now from the three sets of texts obtained, the Text Summarization Model (Modified) is used for each of these sets. This model is little modified to show results such that those sentences which have the most frequently used keywords should appear on top.

E. Compiling the text:

After applying the model to each of these sets, the number of sentences obtained from each of the sets is also stored and an average is calculated. This is used to determine the number of sentences from each set of processed text from the model that will be present in the final description. This average helps ranking the sets in the following order (No of sentence to extract):

1. Processed text from <h1> tag
2. Processed text from Wikipedia
3. Processed text from <h2> and <h3> tag

More the rank is, more the number of sentences it will contribute in the final description

The final compiled text is again check. If its length is more, it is again processed using the modified NLP Model and then presented to user. Else it is directly presented to user.

V. Conclusions

In present work it is clear that not only information but contextually accurate, relevant information is a critical tool for the success of business today. Being able to source relevant information in context to the subject gives ultimate competitive advantage rather than working through traditional, time consuming and iteration approach.

In this project, the text is broken down into almost 50% where this 50% is the abstract or the summary of the complete text. This helps to generate or convert whole paragraph into one third with highly important part as in the form of extract without violating the meaning of paragraph.

Text Summarization, as discussed before, can be used in a lot of fields, from the abstraction of News articles to reducing short books into a short summary. The advantage of this is the ability to reduce the time complexity of user to read these texts by making the text shorter and reducing the efforts. This NLP model with a little modification can also be used to generate SEO descriptions for web developers. This application of Natural Language Processing explains how widely it can be used. This project does quite a good job without the use of Machine Learning algorithms, which produces good results but the actual implementation is very complicated and hardware dependent. Therefore, after comparison, this model turns out to be quite efficient in summarising the text with simple algorithms and some NLP operations.

VI. Acknowledgment

We would like to express our special thanks to our guide Prof. Pravin Khandare and head of our computer department. Prof. Vivek Deshpande who helped us in every possible way and resolved the problems we faced.

We would also like to express our gratitude towards Mr. Saket Goyal for guiding and supporting us.

VII. References

[1] NLTK Documentation: https://www.nltk.org/

[2] Regular Expression Python Documentation: https://docs.python.org/2/howto/regex.html

[3] Beautiful Soup Documentation: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[4] Natural Language Processing – Wikipedia: https://en.wikipedia.org/wiki/Natural_language_processing

[5] Google Index Ranking Factors: https://backlinko.com/google-ranking-factors

[6] Search Engine Optimization: https://en.wikipedia.org/wiki/Search_engine_optimization