Automatic Text Summarisation

[1]Tarun Aggarwal, [2]Vishal Tyagi, [3]Utkarsh Dwivedi, [4]Yash Kumar Sharma
[1]*Assistant Professor, Computer Science, Raj Kumar Goel Institute of Technology, Uttar Pradesh, India*
[2,3,4]*B. Tech, Student, Computer Science, Raj Kumar Goel Institute of Technology, Uttar Pradesh, India*

Abstract - **The objective of this briefing is to propose an application that simplifies the process of extracting meaning from large documents and websites, by making the use of text summarization, i.e., extracting only meaningful excerpts from a large document or website. The Internet is amassed with text data that spans hundreds of pages, sometimes even greater, that requires its meaning to be extracted to both better help discover relevant information and to consume relevant information faster. This briefing proposes an application for the same, in two paradigms i.e., a web-based application, and a Chrome extension. This is an AI based application, using state-of-the-art technologies such as NLP. Two algorithms are proposed for the solution, Abstractive and Extractive Text Summarization; and their results are compared for a sample web page and a document, along with a user-specified web page and document respectively. Along with a dedicated backend occupied by NLP model, a middleware would also be developed as a part of the application, whose sole purpose is the preprocessing of web page as well as documents. At the end of the briefing, the deployment strategies of the application would be discussed, along with the limitations encountered while designing the application. The briefing would be concluded with the future works that are needed with the application, along with the references used in the development of this briefing.**

Index Terms - **AI (Artificial Intelligence), DL (Deep Learning), DRM (Device Rights Management), HTML (Hyper Text Markup Language), HTTP (Hyper Text Transfer Protocol), NLP (Natural Language Processing), NLTK (Natural Language Tool Kit), XML (Extensible Markup Language).**

I. BACKGROUND

There is an enormous amount of textual material, and it is only growing every single day. Think of the internet, consisting of web pages, news articles, status updates, blogs and so much more. The data is unstructured and the best that we can do to navigate it is to use search and skim the results. There is a great need to reduce much of this text data to shorter, focused summaries that capture the salient details, both so we can navigate it more effectively as well as check whether the larger documents contain the information that we are looking for.

II.SUMMARIZATION AND ITS APPLICATIONS

Summarization is the task of condensing a piece of text to a shorter version, reducing the size of the initial text while at the same time preserving key informational elements and the meaning of content. Since manual text summarization is a time expensive and generally laborious task, the automatization of the task is gaining increasing popularity and therefore constitutes a strong motivation for academic research.

There are important applications for text summarization in various NLP related tasks such as text classification, question answering, legal texts summarization, news summarization, and headline generation. Moreover, the generation of summaries can be integrated into these systems as an intermediate stage which helps to reduce the length of the document. In the big data era, there has been an explosion in the amount of text data from a variety of sources. This volume of text is an inestimable source of information and knowledge which needs to be effectively summarized to be useful. This increasing availability of documents has demanded exhaustive research in the NLP area for automatic text summarization. Automatic text summarization is the task of producing a concise and fluent summary without any human help while preserving the meaning of the original text document.

III.CHALLENGES OF TEXT SUMMARIZATION

It is very challenging, because when we as humans summarize a piece of text, we usually read it entirely to develop our understanding, and then write a summary highlighting its main points. Since computers lack human knowledge and language capability, it makes automatic text summarization a very difficult and non-trivial task.

IV.LITERATURE REVIEW

The paper by Chin-Yew Lin [1] entails the introduction of Recall Oriented Understudy for Gisting Evaluation or ROUGE. It is an automatic evaluation package for text summarization, and four different measures of ROUGE are briefed, wiz., ROUGE-N, ROUGE-L, ROUGE-W and ROUGE-S. ROUGE works by measuring the quality of summary generated with that of ideal summaries created by humans. This method finds applications across single document and multi-document summaries.

The paper by Akshil Kumar et al. [2] is an analysis of the performance of three keyword extraction algorithms namely Text Rank, Lex Rank and LSA. Firstly, the paper gives a brief introduction about the different text summarization techniques. These algorithms are explained and implemented in Python. ROUGE-L is used to evaluate the effectiveness of extracted keywords. The results obtained by different algorithms are compared with a human-written summary of the same content. In the end, TextRank algorithm is declared the winner amongst these algorithms.

In the paper by Pankaj Gupta et al. [3], the author has reviewed different techniques of Sentiment analysis and different techniques of text summarization. Sentiment analysis is a machine learning approach in which the machine learns and analyzes the sentiments, emotions present in the text. This paper is a review of different techniques of Sentiment Analysis as well as text summarization [3]. Sentiment analysis is a subsection of NLP, which analyses the emotions present within the text. Traditional classification methods such as Naive Bayes and SVM are employed. Whereas text summarization employs NLP and linguistic features to check the importance of words and sentences that are fit to be included in the final summary. This paper also presents a survey of previous research works done in both Sentiment Analysis and text summarization, so that new research areas can be explored by consideration of merits and demerits of previous works, techniques and strategies.

The paper by Harsha Dave et al. [4] presents an approach to generate abstractive summary from extractive summary using WordNet ontology, for which multiple document formats such as txt, pdf, docx etc. are used. The author first describes various text summarization techniques, which is then followed by step-by-step discussion of multiple document text summarization approaches. These are then compared with a human-generated summary of the same content, which shows the good results obtained by the model.

The paper, by Yihong Gong et al. [5] presents a proposal of two methods that create the generic text summaries by ranking and extracting sentences from main text documents. First method, called Information retrieval, ranks the sentence relevance by providing relevance scores to sentences. Second method uses the LSA technique, which is based on Latent Semantic Indexing, and is used to identify the semantic importance of sentences, for summary creations. Author employs Single Value Decomposition (SVD) to generate a text summary. The author later explains SVD methods in a clear manner, along with the impact of different Weighted schemes on the performance of the summaries. The paper concludes with the comparison of results with human-generated summaries of the same content, and it is shown that the model creates better human-like abstractive summaries. Furthermore, author proposes to investigate various machine learning techniques so as to improve the accuracy.

V.INFERENCES FROM LITERATURE

The daily increase of textual data has made it even more difficult to read the entire material so as to extract necessary information, which is both time-consuming as well as laborious for any human being. Therefore, Automatic Text Summarization becomes more important, which has led to many techniques being discussed in the literature survey. It has many real-world applications such as documents summarization, news and articles summarization, review systems, recommendation systems, social media monitoring and survey responses systems. Above mentioned techniques can be used in a variety of manners, including a mix-and-match approach, which can lead to higher accuracy and lower computational trade-off.

VI.NATURAL LANGUAGE PROCESSING

The power of a computer program to understand human language as it is spoken and written [5]. Whether language is spoken or written, natural language processing uses artificial intelligence to capture real-world input, process it, and make sense of it in a way that a computer can understand. Just as humans have a variety of senses - such as hearing aids and visual aids - computers have programs to read them and microphones to collect sound. And just as people have brains to process that input, so computers have a system in place to process the input properly. At some point in the process, the input is converted to a computer-assisted code.

A. Tokenization

Tokenization is breaking the raw argument into small chunks. Tokenization breaks the raw argument into words, sentences known as tokens. These tokens help in understanding the context or developing the model for the NLP. The tokenization helps in interpreting the meaning of the argument by analyzing the sequence of the words.

There are altered methods and libraries accessible to achieve tokenization. NLTK, Gensim, Keras are some of the libraries that can be acclimated to achieve the task. Tokenization can be done to either abstracted words or sentences. If the argument is breach into words application some break technique it is alleged word tokenization and same break done for sentences is alleged sentence tokenization. Stop words are those words in the argument which does not add any acceptation to the book and their abatement will not affect the processing of argument for the authentic purpose. They are removed from the cant to abate noise and to abate the ambit of the affection set. There are assorted tokenization techniques accessible which can be applicative based on the accent and purpose of modelling.

B. Stop Words

Stop words are the words in any accent which does not add abundant meaning to a sentence. They can cautiously be abandoned without sacrificing the acceptation of the sentence. For some search engines, these are some of the best common, abbreviate function words, such as the, is, at, which, and on. In this case, stop words can account problems back searching for phrases that accommodate them, decidedly in names such as “The Who” or “Take That”.

If we have a task of text classification or sentiment analysis then we should remove stop words as they do not provide any information to our model, i.e., keeping out unwanted words out of our corpus, but if we have the task of language translation then stop words are useful, as they have to be translated along with other words.

C. Stemming and Lemmatization

Stemming is a technique used to abstract the abject form of the words by removing affixes from them. It is aloof like acid down the branches of a timberline to its stems. For example, the stem of the words eating, eats, eaten is eat. Search engines use stemming for indexing the words. That’s why rather than caching all forms of a word, a search engine can store only the stems. In this way, stemming reduces the admeasurement of the basis and increases retrieval accuracy.

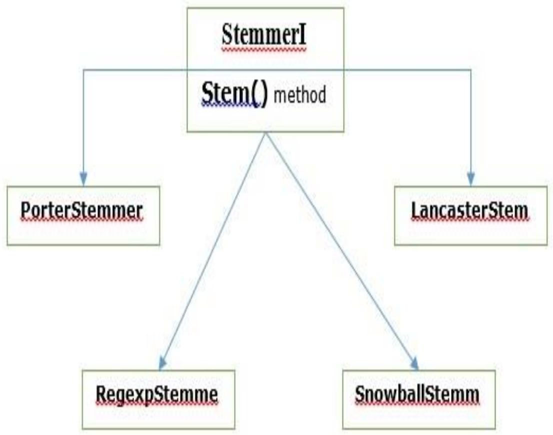

Figure 1: Types of Stemming [7]

Lemmatization method is like stemming. The achievement we will get afterwards lemmatization is alleged ‘lemma’, which is a root word rather than root stem, the achievement of stemming. Afterwards lemmatization, we will be accepting a valid word that agency the aforementioned thing. NLTK provides WordNetLemmatizer class which is a attenuate wrapper about the wordnet corpus. This class uses morphy() function to the WordNet CorpusReader class to acquisition a lemma.

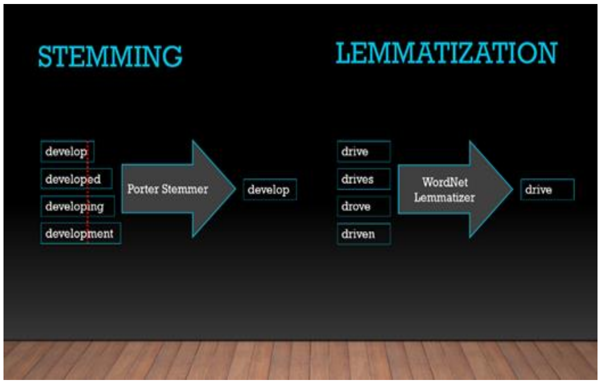

Figure 2: Stemming and Lemmatization [8]

VII.TEXT SUMMARIZATION ALGORITHMS

Text summarization methods can be grouped into two main categories, Extractive and Abstractive methods. Extractive Text Summarization is the traditional method developed first. The main objective is to identify the significant sentences of the text and add them to the summary. You need to note that the summary obtained contains exact sentences from the original text. Abstractive Text Summarization is a more advanced method, many advancements keep coming out frequently). The approach is to identify the important sections, interpret the context and reproduce in a new way. This ensures that the core information is conveyed through shortest text possible. Note that here, the sentences in summary are generated, not just extracted from original text.

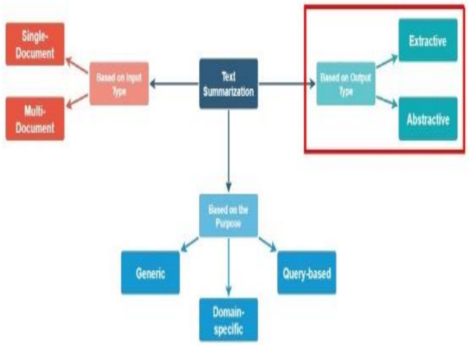

Figure 3: Stages of Text Summarization [9]

A. How to perform Text Summarization

Step 1: First is the split of the paragraph into its corresponding sentences. The best way of doing the conversion is to extract a sentence whenever a period appears.

Step 2: Next is the text processing by removing the stop words (extremely common words with little meaning such as “and” and “the”), numbers, punctuation, and other special characters from the sentences. Performing the filtering assists in removing redundant and insignificant information which may not provide any added value to the text’s meaning.

Step 3: Tokenizing the sentences is done to get all the words present in the sentences.

Step 4: Thereafter is the calculation of the weighted occurrence frequency of all the words. To achieve this, we divide the occurrence frequency of each of the words by the frequency of the most recurrent word in the paragraph.

Step 5: Finally, the substitution of each of the words found in the original sentences with their weighted frequencies. Then, we’ll compute their sum. Since the weighted frequencies of the insignificant words, such as stop words and special characters, which were removed during the processing stage, is zero, it’s not necessary to add them.

VIII.WEB SCRAPPING

Web Scrapping is a technique or a method we use to extract data from the World Wide Web (WWW), or any other documents and save it to a file system or database for later retrieval or analysis. Due to the fact that an enormous or gigantic amount of heterogeneous data is constantly generated every moment, web scrapping is widely acknowledged as an efficient and powerful technique for collecting the data.

Web scrapping can be used for a wide variety of tasks, such as weather data monitoring, price change monitoring, product review collection and if we talk about its uses at macro level then, the metadata of nearly every website is constantly scrapped to build up Internet search engines like Google or Duck Duck Go search engine.

Now a days web scrapping tools are not only capable of parsing markup languages or JSON files but also integrating with computer visual analytics (Butler 2007) and NLP to make the scrap content more summarized with more relevant data.

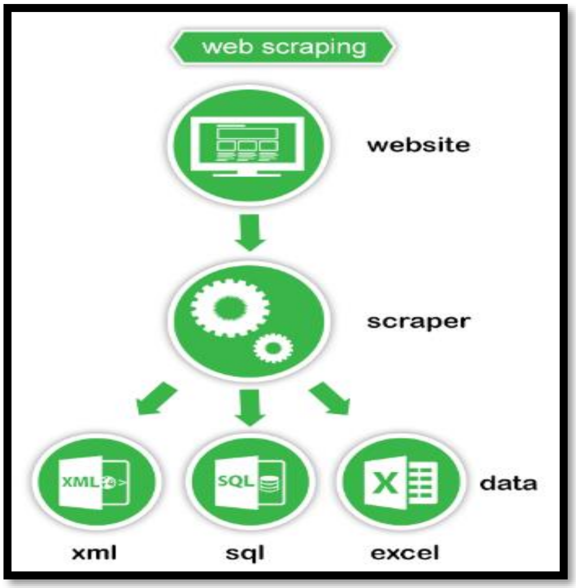

Figure 4: Web Scraping [10]

A. Beautiful Soup

Beautiful Soup is a Python package for extracting data from markup languages such as HTML, XML, and others.

It allows us to extract material from a webpage and store it in the preferred file format by removing HTML markups.

- The Beautiful Soup library helps with isolating titles and links from webpage by identifying the parsers and markup tags.
- It can extract or collect all of the text from HTML tags.

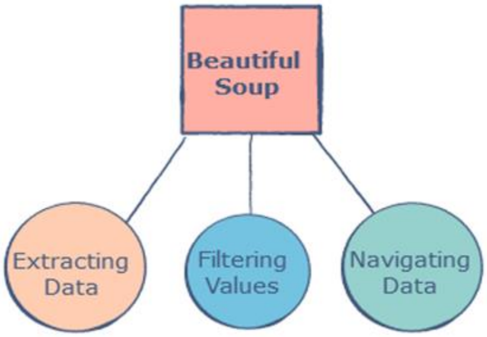

Figure 5: Stages of Beautiful Soup [11]

- And it alters the HTML in the document with which we’re working, so that if in future we need updated data we can easily extract it from the websites.

Some key features of beautiful soup are:

- Beautiful Soup provides Pythonic idioms for navigating, searching, and modifying a parse tree.
- Beautiful Soup automatically converts incoming documents to Unicode and outgoing documents to UTF-8.
- Beautiful Soup allows us to try out different parsing strategies or trade speed for flexibility.

IX.BROWSER EXTENSION

A browser extension is a small piece of software that enhances the functionality or capacity of a web browser. A browser extension, commonly known as a plug-in, can use the same Application Program Interfaces (APIs) as JavaScript can on a web page, but it can also access its own set of APIs, allowing it to do more.

While extensions are most commonly used to add features and improve the functioning of a website, they can also be used to eliminate unpleasant website elements like pop-up advertisements and auto-play features for online videos.

X.REST API

REST stands for Representational State Transfer and an Application Programming Interface (API) are a set of definitions and protocols for creating and integrating software applications.

A REST API (also known as a RESTful API) is a type of application programming interface (API or web API) that adheres to the REST architectural style's limitations and allows interaction with RESTful web services.

In order for an API to be considered RESTful, it's to adapt to those criteria:

1. A client-server architecture made of clients, servers, and resources, with requests managed through HTTP.
2. Stateless client-server communication, meaning no client information is stored between get requests and every request is separate and unconnected.
3. Cacheable data that streamlines client-server interactions.

4. A uniform interface between components in order that information is transferred in an exceedingly standard form. this needs that:
   - resources requested are identifiable and break away the representations sent to the client.
   - resources will be manipulated by the client via the representation they receive because the representation contains enough information to try and do so.
   - self-descriptive messages returned to the client have enough information to explain how the client should process it.
   - hypertext/hypermedia is on the market, meaning that after accessing a resource the client should be ready to use hyperlinks to search out all other currently available actions they'll take.
5. A layered system that organizes each sort of server (those accountable for security, load-balancing, etc.) involved the retrieval of requested information into hierarchies, invisible to the client.
6. Code-on-demand (optional): the flexibility to send executable code from the server to the client when requested, extending client functionality.

Though the RESTAPI has these criteria to evolve to, it's still considered easier to use than a prescribed protocol like SOAP (Simple Object Access Protocol), which has specific requirements. In contrast, REST could be a set of guidelines which will be implemented as required, making REST APIs faster and more lightweight, with increased scalability—perfect for Internet of Things (IoT) and mobile app development.

XI.LIMITATIONS

As with any foundational technology or project, this one too has inevitable limitations. The most trivial and profound is the access of DRM-protected content from such websites for summarization. Access of such content is protected by law and therefore, must be handled carefully. Another limitation is the accuracy of AI-based backend, which can be improved by using technologies such as DL etc. Another limitation is the fact thar, since the software comes packed in a Chrome extension, it supports only Chromium-based browsers such as Google Chrome, Microsoft Edge (New Version) etc. This signifies that the software is unusable in popular browsers such as Mozilla Firefox and Apple Safari.

XII.CONCLUSION

This briefing proposes an application that simplifies the process of extracting meaning from large documents and websites, by making the use of text summarization, i.e., extracting only meaningful excerpts from a large document or website. The briefing provides information on all the technologies required to create such software, along with the necessary algorithms to achieve maximum accuracy. Along with a dedicated backend occupied by NLP model, a middleware is also proposed as a part of the application, whose sole purpose is the preprocessing of web page as well as documents. At the end of the briefing, the deployment strategies of the application are discussed, along with the limitations encountered while designing the application.

XIII.FUTURE SCOPE

This briefing proposed an application that simplifies the process of extracting meaning from large documents and websites, by making the use of text summarization, i.e., extracting only meaningful excerpts from a large document or website. There are websites that enforce their security by the use of DRM and scrapping such websites yield no results [6]. Therefore, the next version of this application may work on such websites for summarization. Further versions may also include the upload of different kinds of document extensions such as pdf, docx etc. for efficient summarization, since most of the people possess such documents, which are in dire need of summarization.

REFERENCES

[1] Chin-Yew Lin, “Rouge: A Package for Automatic Evaluation of Summaries.” Barcelona Spain, Workshop o Text Summarization Branches Out, Post- Conference Workshop of ACL 2004.

[2] Akshil Kumar, Aditi Sharma, Siddhant Sharma, Shashwat Kashyap, “Performance Analysis of Keyword Extraction Algorithms Assessing Extractive Text Summarization.” International Conference on Computer, Communication, and Electronics (Comptelix), 2017.

[3] Pankaj Gupta, Ritu Tiwari and Nirmal Robert, “Sentiment Analysis and Text Summarization of Online Reviews: A Survey.” International Conference on Communication and Signal Processing, 2016.

[4] Harsha Dave, Shree Jaiswal, “Multiple Text Document Summarization System using Hybrid Summarization Technique.” 1st International Conference on Next Generation Computing Technology (NGCT), 2015.

[5] Vishal Tyagi, Shraddha Singh. “Sentiment Analysis to Detect Mental Depression Based on Twitter Data", Volume 8, Issue IX, International Journal for Research in Applied Science and Engineering Technology (IJRASET) Page No: 268-274, ISSN: 2321-9653.

[6] Chin-Yew Lin and Eduard Hovy, “The potential and Limitations of Automatic Sentence Extraction for Summarization”.

[7] Types of Stemming https://www.tutorialspoint.com/natural_language_toolkit/natural_language_toolkit_stemming_lemmatization.htm

[8] Stemming and Lemmatization https://www.kaggle.com/getting-started/186152

[9] Stages of Text Summarization https://hackernoon.com/summarization-with-wine-reviews-using-spacy-b49f18399577

[10] Web Scraping https://www.unescap.org/sites/default/files/Leveraging_online_price_data_from_web_crawling_Malaysia_Stats_Cafe_30Nov2020.pdf

[11] Stages of BeautifulSoup https://medium.com/@belen.sanchez27/5-steps-to-get-started-with-web-scraping-using-beautiful-soup-8d954a406627