SPLITBRAIN
(Text Summarization)

J COMPONENT PROJECT REPORT

Winter 2019-20

Submitted by

Raghav Jindal (18BCE2080)

Kunal Agarwal (18BCE2108)

in partial fulfillment for the award of the degree of **B. Tech**

in

Computer Science and Engineering

Vellore-632014, Tamil Nadu, India

School of Computer Science and Engineering

October, 2020

1. Title

SplitBrain (Text Summarization)

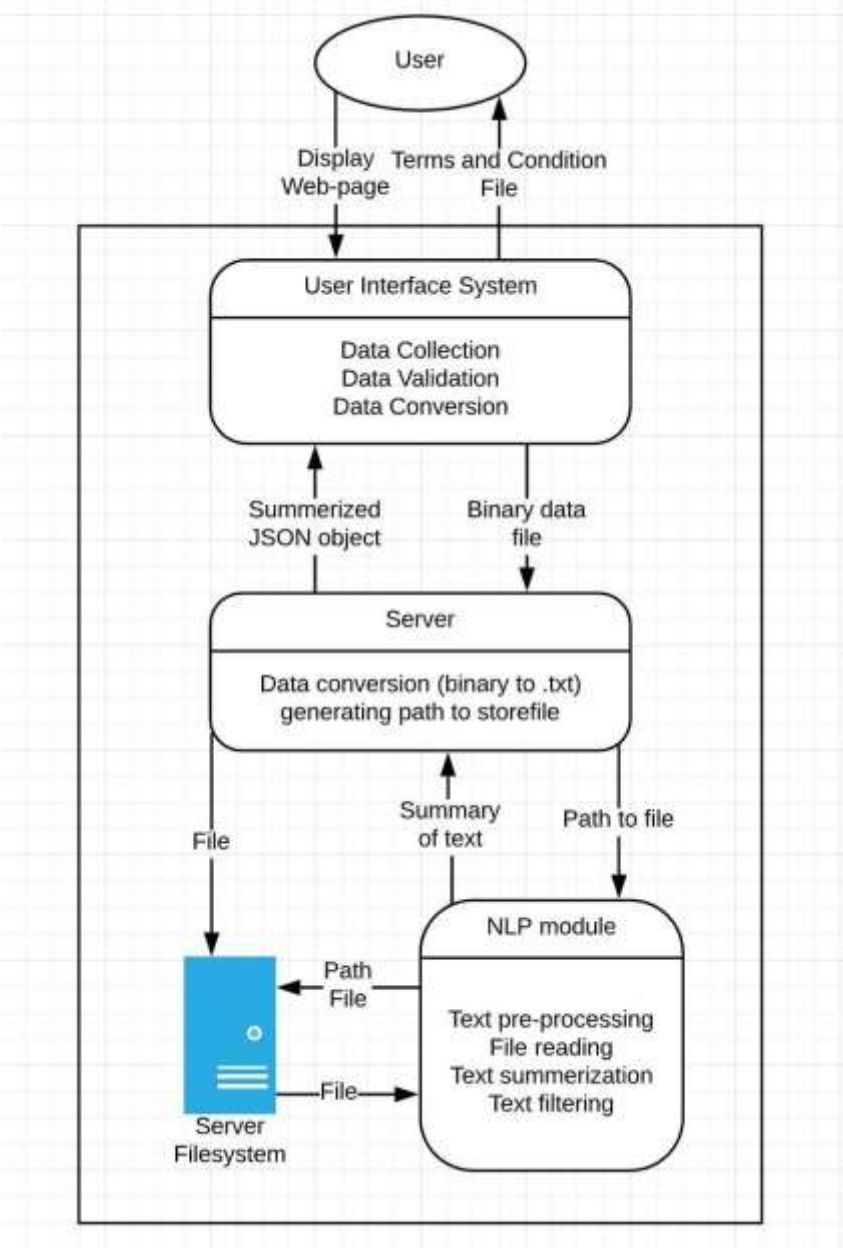
2. Abstract

Text summarization is the technique to shorten long pieces of text, with the intention to create a meaningful and fluent gist having only the key points outlined in the document. Text summarization is a well known problem in machine learning and natural language processing, in the last few decades, text summarization has gotten an exponential level of importance due to the fact that enormous amount of data is online, and it has the potential to take out useful information that could be handled relatively easily by humans and could be used for a wide range of purposes, such as text assessment. This report we try to presents an automatic process for text summarization that relies on TextRank Algorithm to find the most relevant and important information in the input text. The automatic produced brief summary of the texts are compared with summaries created by the domain experts. Our method summarizes text by using Extractive Summarization and Abstractive Summarization for text summarization. The proposed summarization method has been trained and tested in experiments using a dataset of English texts provided by our fellow students of NLP course. At the end, the proposed approach was compared with other methods including a naive baseline, Score, Model and Sentence, using ROUGE measures.

3. Introduction

Text Summarization is a very helpful tool of Natural Language Processing which has huge impact on our day to day lives. With the on-growing digital media and ever growing publications – who on earth has the time and patience to go through all the articles / files / books to decide whether any of them are useful or not? Thankfully – the technology is already here. Have you come across mobile apps such as inshorts? It's an innovative news app which is designed to convert news articles into a 50-60-word summary. And that is exactly what we are going to do — Text Summarization.Text Summarization is arguably one of the most challenging and captivating problems in the domain of Natural Language Processing (NLP). It's the process of generating a brief and meaningful synopsis of text from various text sources such as books, news articles, blog posts, research papers, emails, and tweets. The demand for text summarization systems is peaking these days due to the availability of enormous amounts of textual data. Through this report, we will try to explain the realms of text summarization. We will try to explain how the TextRank algorithm works, and will also implement it in Python.

4. Architecture diagram



5. Background study

Due to the high paced development of the Internet, the volume of data resources is increasingly exponentially. As a huge text data, the way to effectively and precisely retrieve the content of the article has been an intriguing filed of research for data scientists. By analyzing a paragraph or an article, the most common way to start is to proceed with the keywords. The keywords can not just condense the meaning of the text, but can also decipher the main content and notion of

the whole written text. As result, precise and quick extraction of keywords is very important for clustering the text, text summarization and information retrieval.

In past few decades, researchers have carried out a lot of work in this field of keywords extraction technology, by which they've put forward many of keywords searching algorithms, the major algorithms are keywords extraction based on implied subject model (LDA) [2], keywords extraction based on TF-IDF word frequency statistics [9] and keywords extractionbased on word graph model (TextRank) [10]. The three mentioned algorithms have been widely used for their simplicity and precision.

Among the algorithms, the keywords extraction algorithm based on the implied topic model finds the importance of words according to the uniqueness of the topic distribution of text and words ,as this technique usually needs to train the corpus content to get precise information, the quality of keywords drawn out by this method is greatly affected by the topic distribution of the training corpus.

TextRank algorithm is a graph based text sorting algorithm, which uses the relationship between local words to sort the following keywords and draw out the keywords directly from the text itself. TextRank algorithm has the advantages of unsupervised, simple implementation, weak language correlation, and is suitable for all kinds of text processing. But this technique does not look into if the the significance of different words will affect the problem of the nearby nodes weight transfer, and doesn't take into account the whole information document corpus. The weight of words information has no actual meaning, and cannot distinguish between the strength of the connection. The relationship between the words can only be determined by the use of a sliding window within a single sentence, without considering the context as a whole.

To further improve the effect and quality of keywords extraction, many researchers have optimized the above algorithms. Lang Dongdong [8] et al. proposed a key phrase extraction method based on LDA and TextRank. Gu Yijun [3]et al. combined LDA with TextRank, so that the significance of the candidate node words was non-uniformly transferred according to the document set topic distribution. However, the results are mostly affected by the subject distribution of training corpus. Zhang Jin et al. [6], Xie Jin et al. [6] considered improving TFIDF weight based on word position and word span, or using semantic coherence and combining word frequency and position characteristics to carry outweighting [11].

Some researchers introduced the method of information entropy [4].However, these techniques all have some complexity problems or restrictions in the type of articles and corpus size.

Some researchers merged the comprehensive information in the text with the technique of citing news category factors [4,13,16],and added other characteristic factors to weight, which could correct the word frequency dependence problem to a certain limit. However, this method does not look on the influence of the part of speech and the different coverage of keywords.

Biswas S K [1] et al., Yan Ying [13] et al. proposed a keywords extraction method based on graph, which looked into the context, location, centrality, part of speech and other

characteristics of the words respectively, modified the initial weight of the words, etc., and achieved a good extraction effect.

From all the research work in the end we can infer that keywords in a text can not only summarize the theme of the article, but also draw out the main content and notion expressed in the whole article. The results of keyword extraction by the algorithm need to reflect the theme content of the article little bit accurately. As result, accurate and quick extraction of keywords is very important for text clustering, text summarization and information retrieval.

6. Methodology

Our software is divided into three Modules,

1. The User Interface module, or the front end module
2. The server module, or the back end module
3. The NLP module

The user interacts with the front end module, where they upload a .txt file containing the terms and conditions they want to summarise. This file is validated by the module itself to check for the correct file extension, and to check that the file doesn't exceed the maximum size allowed. Once validated, it's converted to a binary format to make it ready for transfer. And is sent to the server module using HTTP protocol, as a part of the POST request. The server module, upon receiving the file, converts it back to a .txt file, and stores it on the disk, with the time of reception of file as the file name, so as to allow differentiation of files. The file is stored and the path to the file is generated, which is further sent to the NLP module as an argument. The NLP module receives the path, retrieving and reading the contents of the file.This content is finally processed upon by our model, and is returned to the server module as an array containing the important bullet points. This array is sent back to the User Interface module as a JSON object, where upon receiving the object the server renders these bullet points onto the screen.

We have used TextRank algorithm in our project, before understanding the TextRank algorithm it is wise to understand PageRank algorithm on which the TextRank algorithm is based.

PageRank is a widely used algorithm by Google Search to rank websites results. PageRank is an algorithm for measuring the importance of web pages and ranking them accordingly. The outputs of PageRank algorithm is a probability distribution used to discover the likelihood that a user randomly clicking on links will arrive at any particular webpage. PageRank is calculated for collections of documents of all sizes.

TextRank algorithm is similar to the PageRank algorithm in the following ways:

• In place of web pages, TextRank use sentences

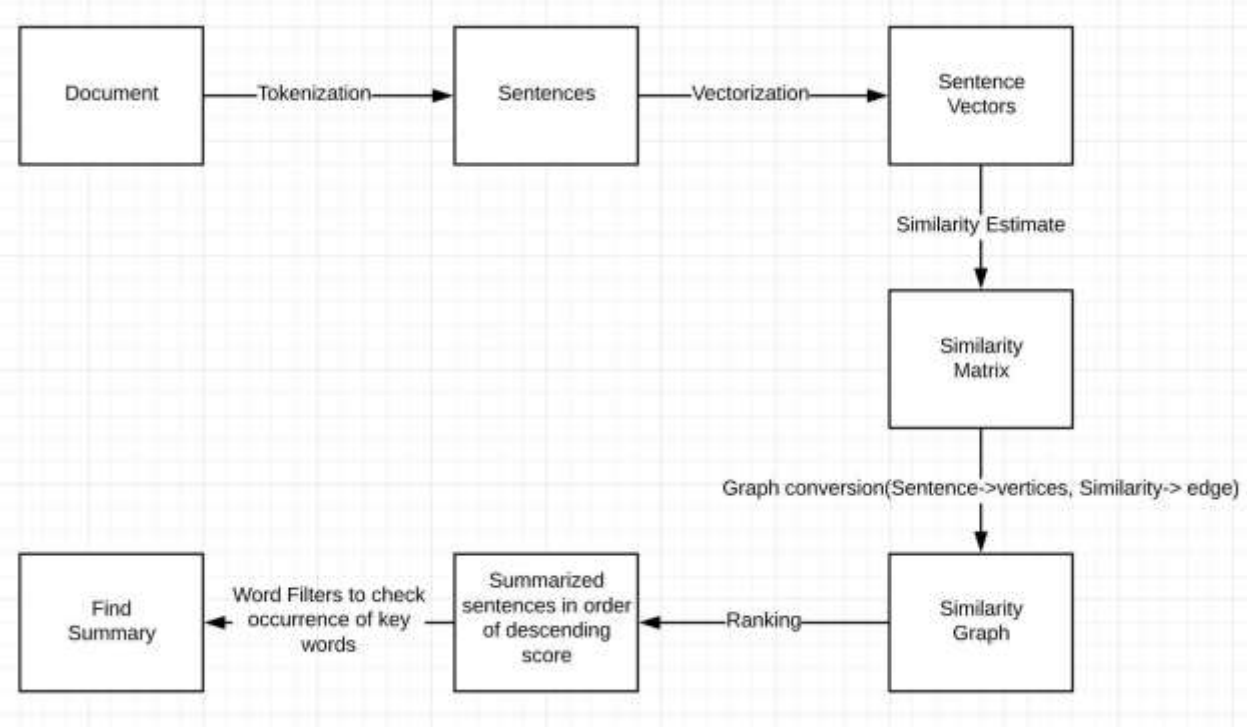
- Similarity in any two sentences is used as an equivalent to the web page transition probability
- Similarity scores are then stored in a square matrix, same as the matrix M used for PageRank

TextRank is extractive and unsupervised text summarization technique.

The first step of it is to concatenate all the text contained in the articles, then split the text into individual sentences, then we will find vector representation for each and every sentence.

Similarities between the sentence vectors are then calculated and stored in a matrix. The similarity matrix is then further converted into a graph, with the sentences as vertices and similarity scores as the edges, for sentence rank calculation, finally a certain number of topranked sentences form the final summary.

7. Proposed model



In our model,

The first step would be to concatenate all the text contained in the document.

Then split the text into individual sentences.

In the next step, we will find vector representation (word embeddings) for each and every sentence using GloVe word embeddings are vector representation of words. We could have also used the Bag-of-Words or TF-IDF approaches to create features for our sentences, but these methods ignore the order of the words

Similarities between sentence vectors are then calculated and stored in a matrix.

The similarity matrix is then converted into a graph, with sentences as vertices and similarity scores as edges, for sentence rank calculation

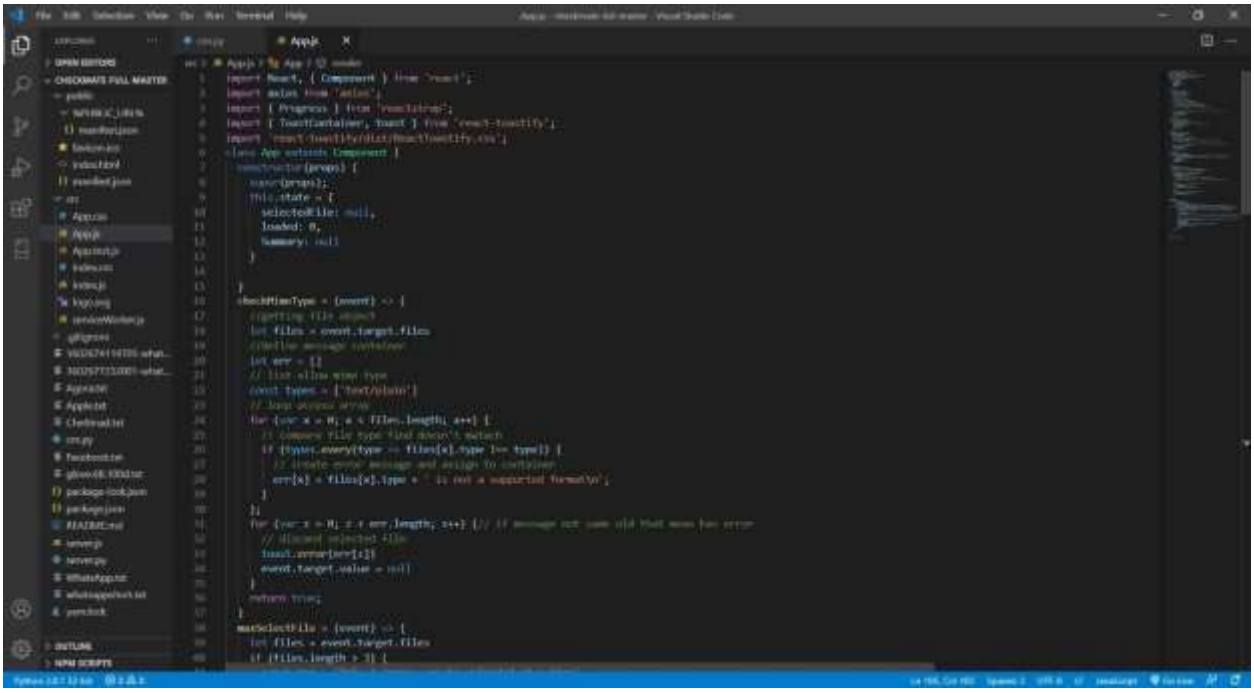
Finally, a filter is applied to check for occurrence of trigger words that may be of importance.

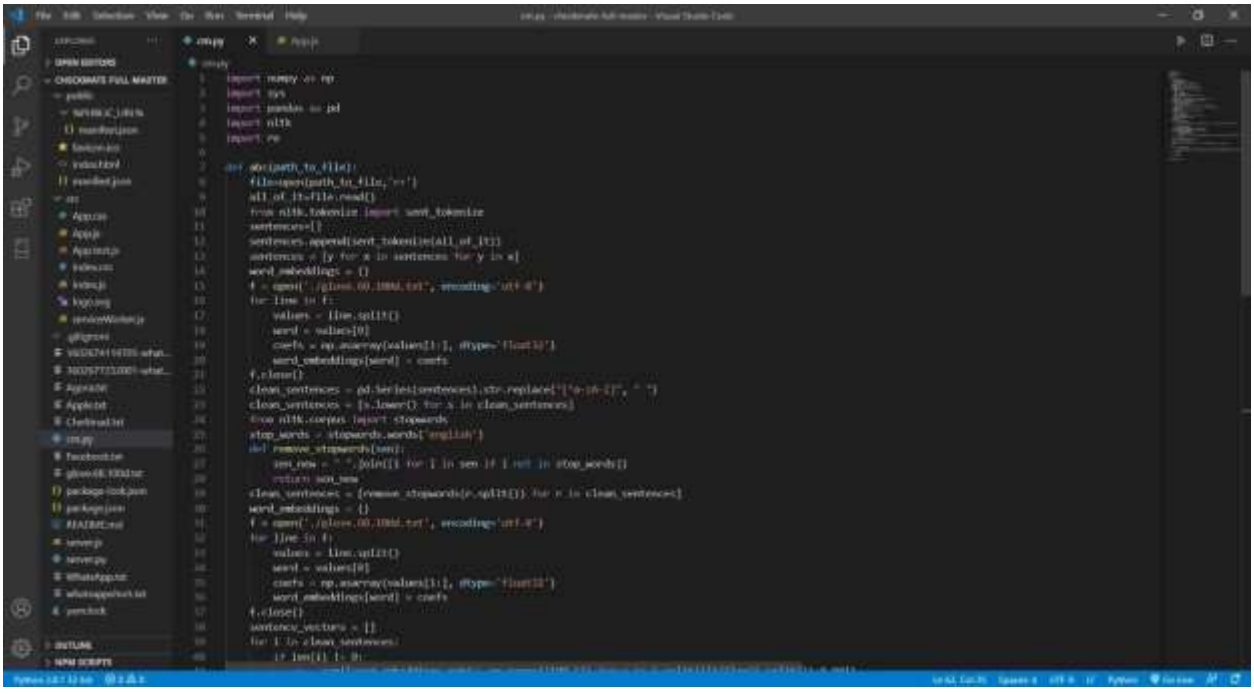
The words are:

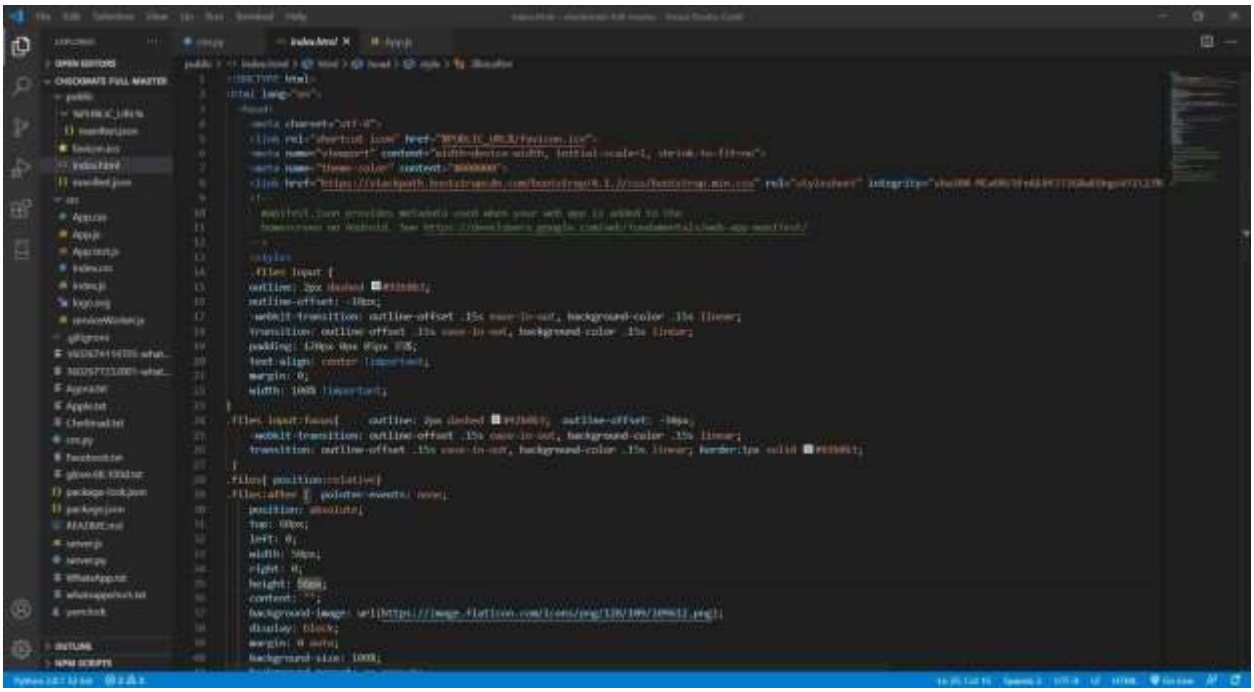
1. Service
2. Privacy
3. Security
4. Data
5. Protect
6. Connect
7. Retain
8. Use
9. Train

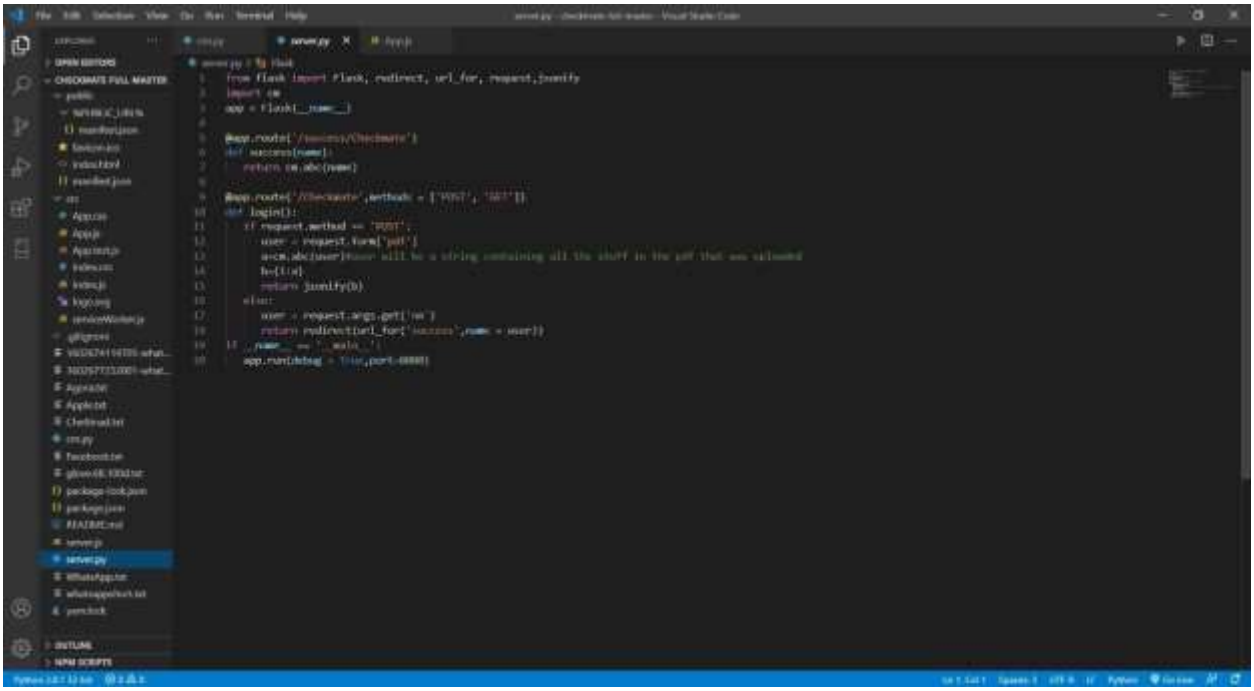
The words are picked for data privacy domain, but can be changed to fit other domains as well.

8. Results and Discussion

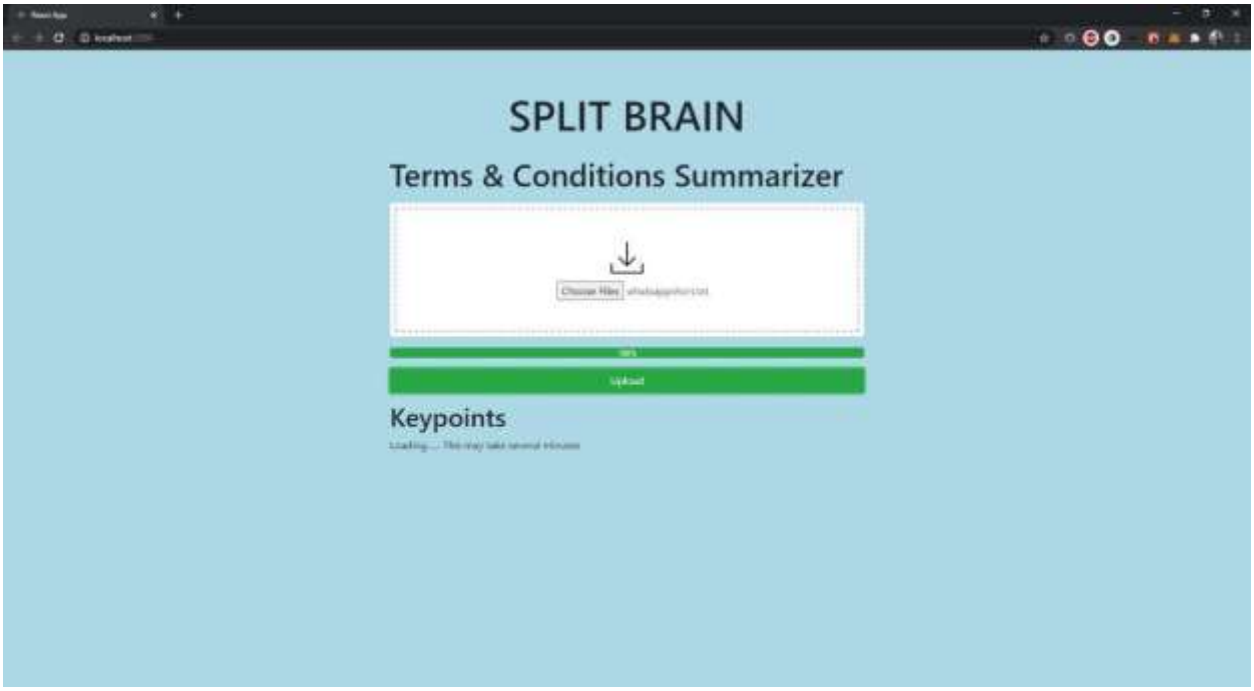


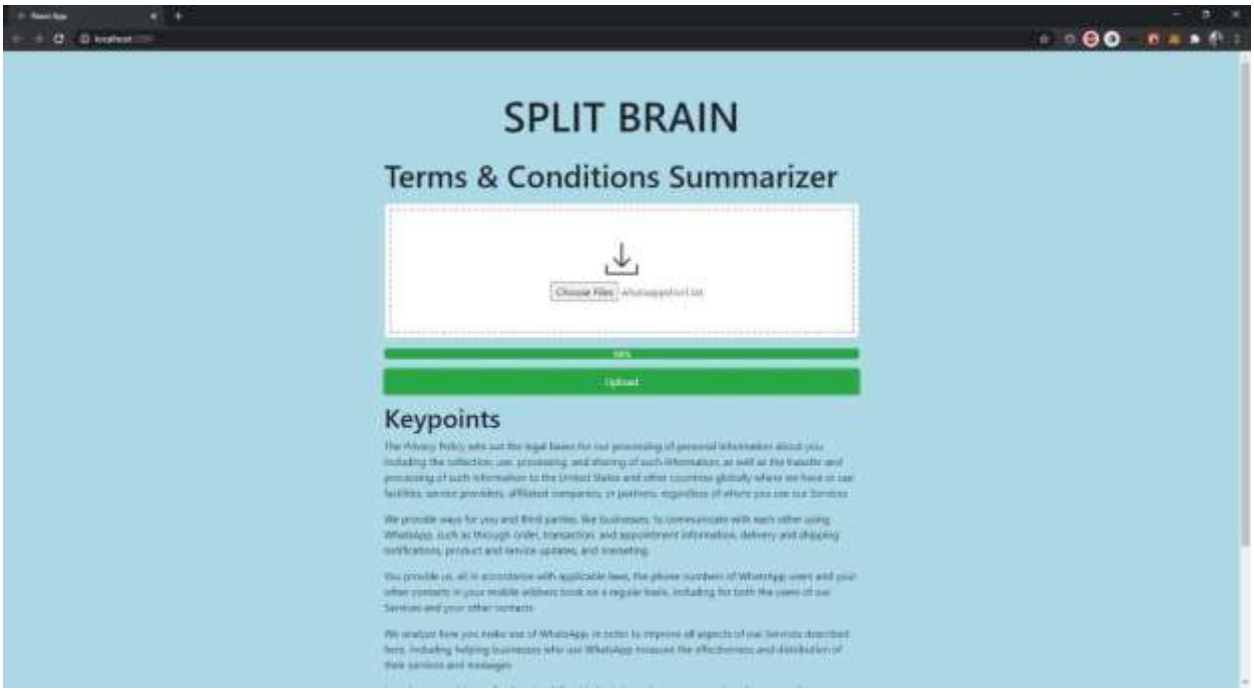







SPLIT BRAIN
Terms & Conditions Summarizer


SPLIT BRAIN
Terms & Conditions Summarizer
Keypoints



9. Conclusion

In this report, we have discussed the method of abstractive summarization as well as extractive summarization. The results of applying extractive summarization proved out to be very effective.
The algorithm does not handle anaphora resolution problem. It is possible that more than one same type of sentences with high score is selected for the summary. To avoid this kind of duplication, we could consider a clustering algorithm and combine it with the our algorithm. As a betterment of this report, we will compare our method with many more existing effective text summarization methods with different performance measurement metrics.

10. References

Approach of Automatic Keyphrase Extraction.Procedia Computer Science 107 (2017), 248–255.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichletallocation. Journal of machine Learning research 3, 01 (2003), 993–1022.

Dongdong Lang, Chenchen Liu, Xupeng Feng, Lijun Liu, and QingsongHuang. 2018. design and implemention of a key phrases extractionscheme in the text based on lda and textrank. Computer Applicationsand Software 03 (2018), 54–60.

H. U. Xue-Gang, L. I. Xing-Hua, Fei Xie, and W. U. Xin-Dong. 2010.Keyword Extraction Based on Lexical Chains for Chinese News WebPages. Pattern Recognition and Artificial Intelligence 123, 01 (2010),45–51.

Juanzi Li, Kuo Zhang, et al.2007. Keyword extraction based on tf/idffor Chinese news document. Wuhan University Journal of NaturalSciences 12, 05 (2007), 917–921.

Li Hang, Chaolan Tang, Yang Xian, and Wanting Shen. 2017. Tex-tRank Keyword Extraction Based on Multi Feature Fusion. Journal ofIntelligence 36, 08 (2017), 183–187.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text.In Proceedings of the 2004 conference on empirical methods in naturallanguage processing.

Saroj Kr. Biswas, Monali Bordoloi, and Jacob Shreya. 2018. A graphbased keyword extraction model using collective node weight. ExpertSystems with Applications 97 (2018), 51– 59.

Xie Jin. 2012. A Method of Automatic Keyword Extraction Based onWord Span. Modern Property Management 11, 04 (2012), 108–111.

Y. Wang, C. Mao, Z. Yu, J. Guo, and L. Luo. 2016. Approach for topicalsentence of news events extraction based on graph. 40, 04 (2016),438–443.

Yan Ying, Qingping Tan, Qinzheng Xie, Zeng Ping, and Panpan Li.2017. A Graph-based

Yang Yan. 2011. Exploration and improvement in keyword extractionfor news based on TFIDF. 13Z (2011), 3551–3556.

Yijun Gu and Tian Xia. 2014. Study on Keyword Extraction withLDA and TextRank Combination. New Te chnology of Library andInformation Service 30, zl (2014), 41–47.

Zhang Jin. 2014. A Method of Intelligence Key Words Extraction Basedon Improved TF-IDF. Journal of Intelligence 33, 04 (2014), 153–155.