TEXT SUMMARIZATION OF BBC NEWS

December 12th ,2022

Harsha Teegala, MS Data Science

1. **INTRODUCTION**

We see long pieces of text information that can be found on news websites, blogs, consumer review websites, and so on. When displaying an output many times the management requires concise report of the tasks and results achieved. It’s a complex process to convert some large chunk of text to understandable and brief format. Natural language processing (NLP) is a branch of computer science, artificial intelligence, and linguistics dealing with computer-human interaction. Text summarization is one of the uses of NLP. It is because of the abundance of data accessible today; text summarization has become critical for extracting precisely the correct amount of information from large texts. Text summarization is a process of reducing the size of original document and producing a summary by retaining important information of original document.1

There are several uses for text summarization; for example, researchers require a tool to produce summaries so they can decide whether to read the complete material or not. Text summarization may also be used to summarize information that users have looked for online. Multi document summarizing can be used by news organizations to group and summarize information from many sources.1 The goal of my project is to summarize news articles of BBC news from different domains using Extractive method and seq2seq.

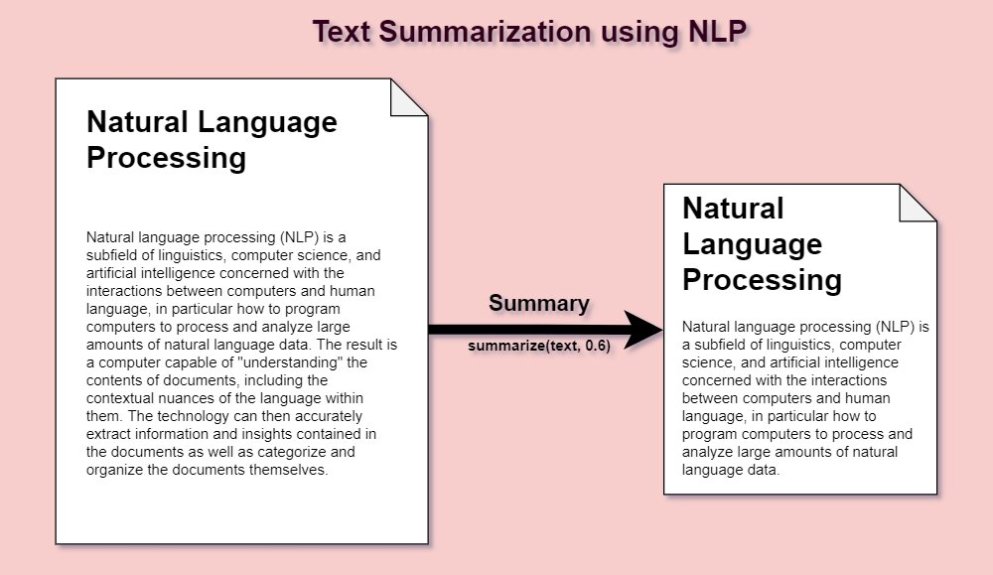


Image Source:  Types of Text Summarization: Extractive and Abstractive Summarization Basics - Turbolab Technologies. Turbolab Technologies. Published September 20, 2021. Accessed December 11, 2022. https://turbolab.in/types-of-text-summarization-extractive-and-abstractive-summarization-basics/

1. **PROBLEM DEFINITION**

The main objective of this project is to process the articles with the given summaries provided by experts for the respective articles and try to predict those summaries using various algorithms. Though before jumping directly to training and testing the models some of the main steps or problem statements are as follows:

1. What type of text summarization method to use, as there are many available techniques
2. What type of data cleaning is required?
3. What feature engineering method(s) works best for the problem statement?
4. Which algorithm should be used and how to check it’s acuuracy?
5. **BACKGROUND**

Abstractive and extractive summarization are the two main methods of summarization.

Abstractive Summarization: Summarization that is abstractive chooses terms based on semantic understanding, even when those words do not occur in the source materials. It seeks to create significant content in a novel manner. To create a new, shorter text that captures the essence of the original material, they analyse and interpret the text using cutting-edge natural language processing algorithms. It is comparable to how people read text articles or blog posts before summarizing it in their own words. 2

***Input document → understand context → semantics → create own summary.***

Extractive Summarization: Extractive approaches aim to condense lengthy texts by choosing a small group of words that capture the key ideas. With this method, the significant parts of sentences are given more weight and used to create the summary. The sentences are ranked further based on their relevance and resemblance to one another using various algorithms and methodologies.2

***Input document → sentences similarity → weight sentences → select sentences with higher rank.***

Text Rank Algorithm is the most used method for extractive summarization. Let’s take a look at the flow of the TextRank algorithm that we will be following:

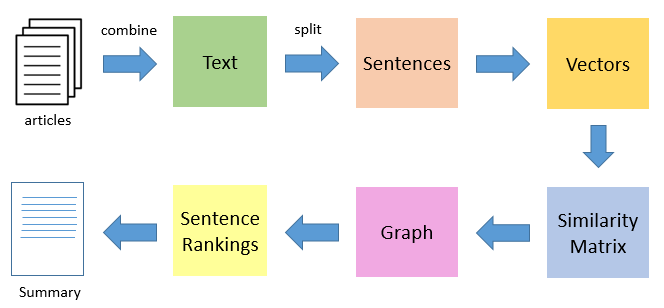


Image Source: Joshi P. Automatic Text Summarization Using TextRank Algorithm. Analytics Vidhya. Published November 1, 2018. Accessed December 12, 2022. https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/

The overview of the algorithm is as follow:

* Concatenating all of the articles' text would be the initial stage, after which the text would be divided up into separate sentences.
* We will locate vector representations (word embeddings) for each and every phrase in the next phase.
* Then, a matrix is created that contains the results of the comparison of sentence vectors.
* For the purpose of calculating sentence rank, the similarity matrix is subsequently transformed into a graph with sentences as nodes and similarity scores as edges.
* The final summary is composed of a specified amount of the top-ranked sentences.5

Purely extractive summaries often times give better results compared to automatic abstractive summaries. This is because of the fact that abstractive summarization methods cope with problems such as semantic representation,  
inference and natural language generation which is relatively harder than data-driven approaches such as sentence extraction.2

Another approach is to use Seq2seq. Our goal is to create a text summarizer that produces a brief summary from a long string of words (in a text body) (which is a sequence as well). This may therefore be modelled as a Many-to-Many Seq2Seq issue.3A typical Seq2Seq model architecture is shown below:

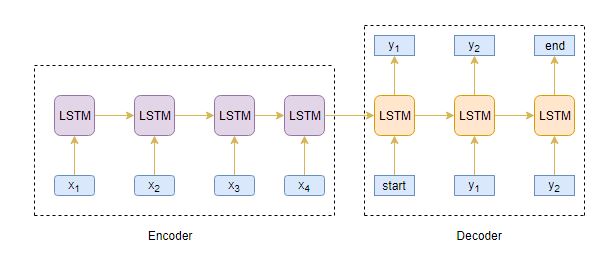
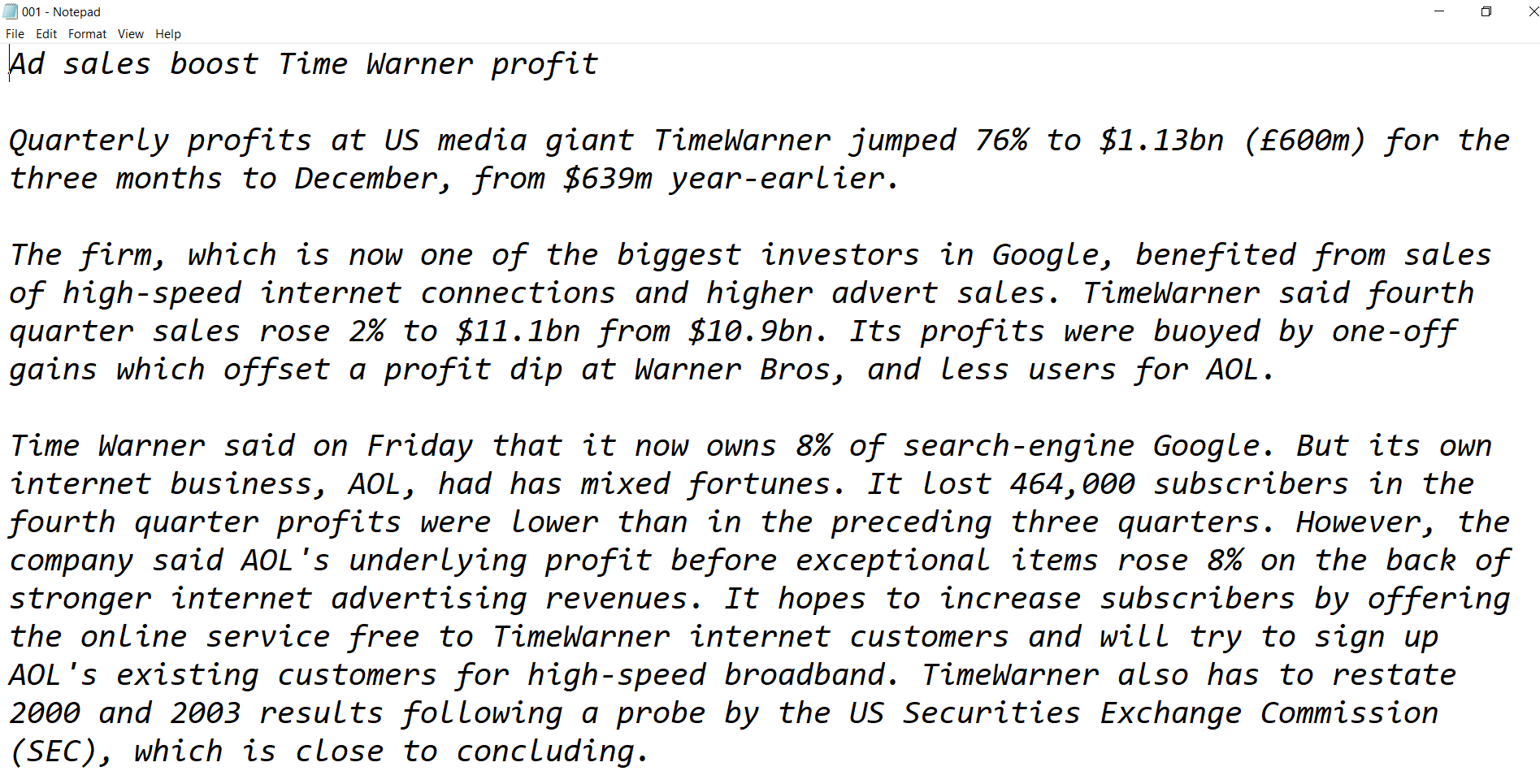


Image Source: Pai A. Text Summarization | Text Summarization Using Deep Learning. Analytics Vidhya. Published June 10, 2019. Accessed December 11, 2022. https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarization-using-deep-learning-python/

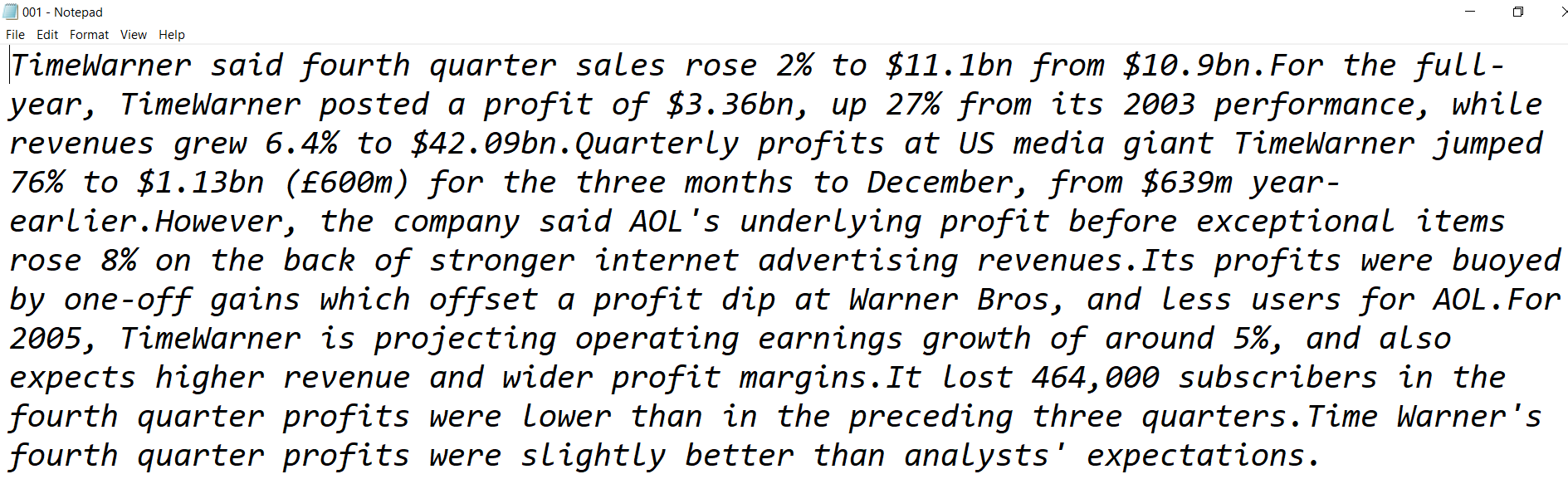
1. **DESCRIBING THE DATASET**

Four hundred seventeen political news stories from the BBC from 2004 to 2005 are included in this dataset for extractive text summarization in the News Articles folder. The Summaries folder has five summaries for each article. The title of each article appears in the first clause. This dataset was made using a dataset for data classification from the 2004–2005 work by D. Greene and P. Cunningham, which consisted of 2225 documents from the BBC news website relating to stories in five thematic categories. The BBC owns all rights, including copyright, in the material of the original papers published in "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering", Proc. ICML 2006.4

I acquired this dataset from Kaggle. It has 2 directories. One for the original text files for articles which includes articles from business, entertainment, politics, sports and tech domain. Similarly, second directory is for Samples of the respective article text files from all sectors. There are total 4450 files. An instance is shown below:



Article from business

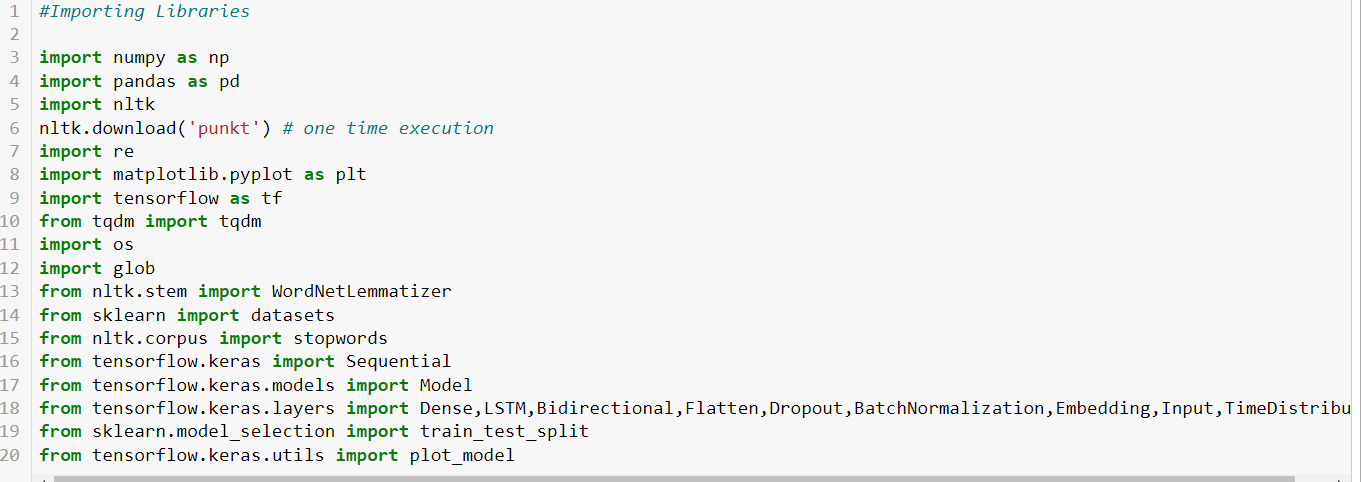


The summary for the above article

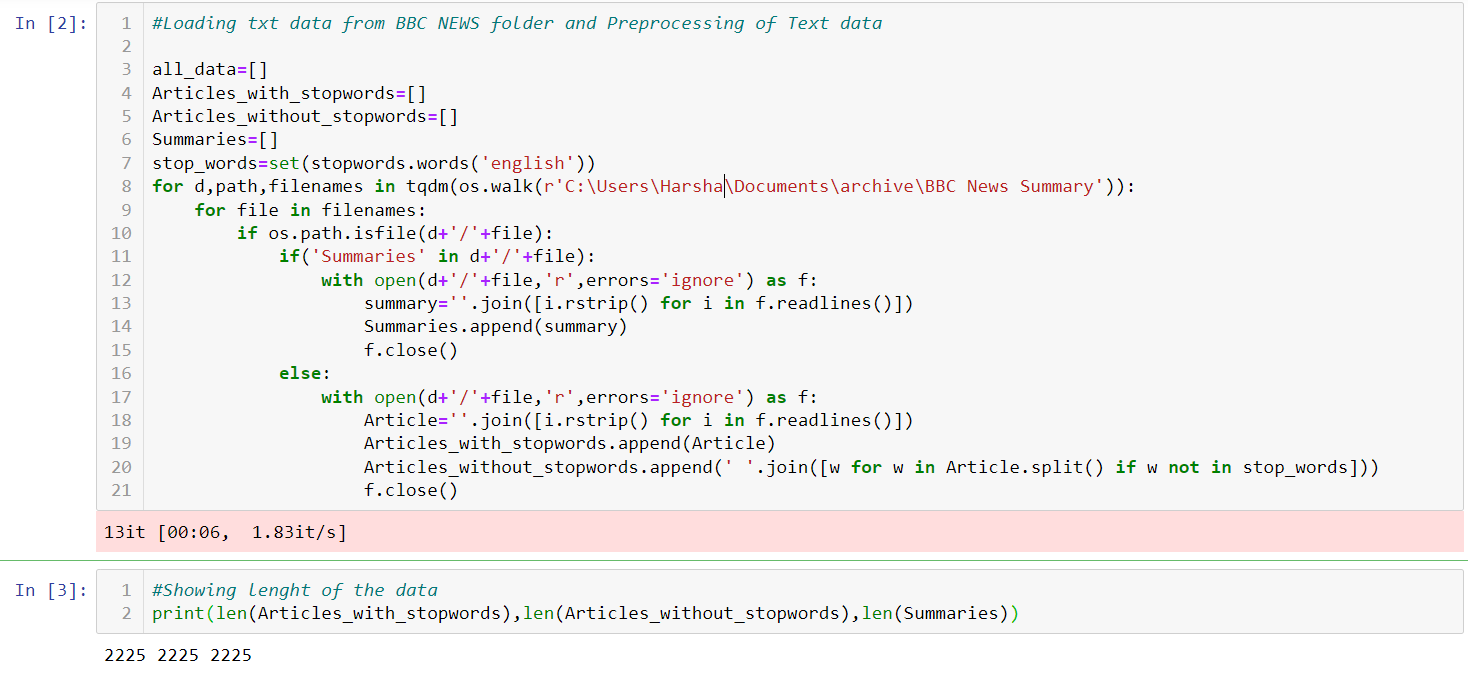
1. **NLP METHODS USED**

The whole project was divided into multiple tasks:

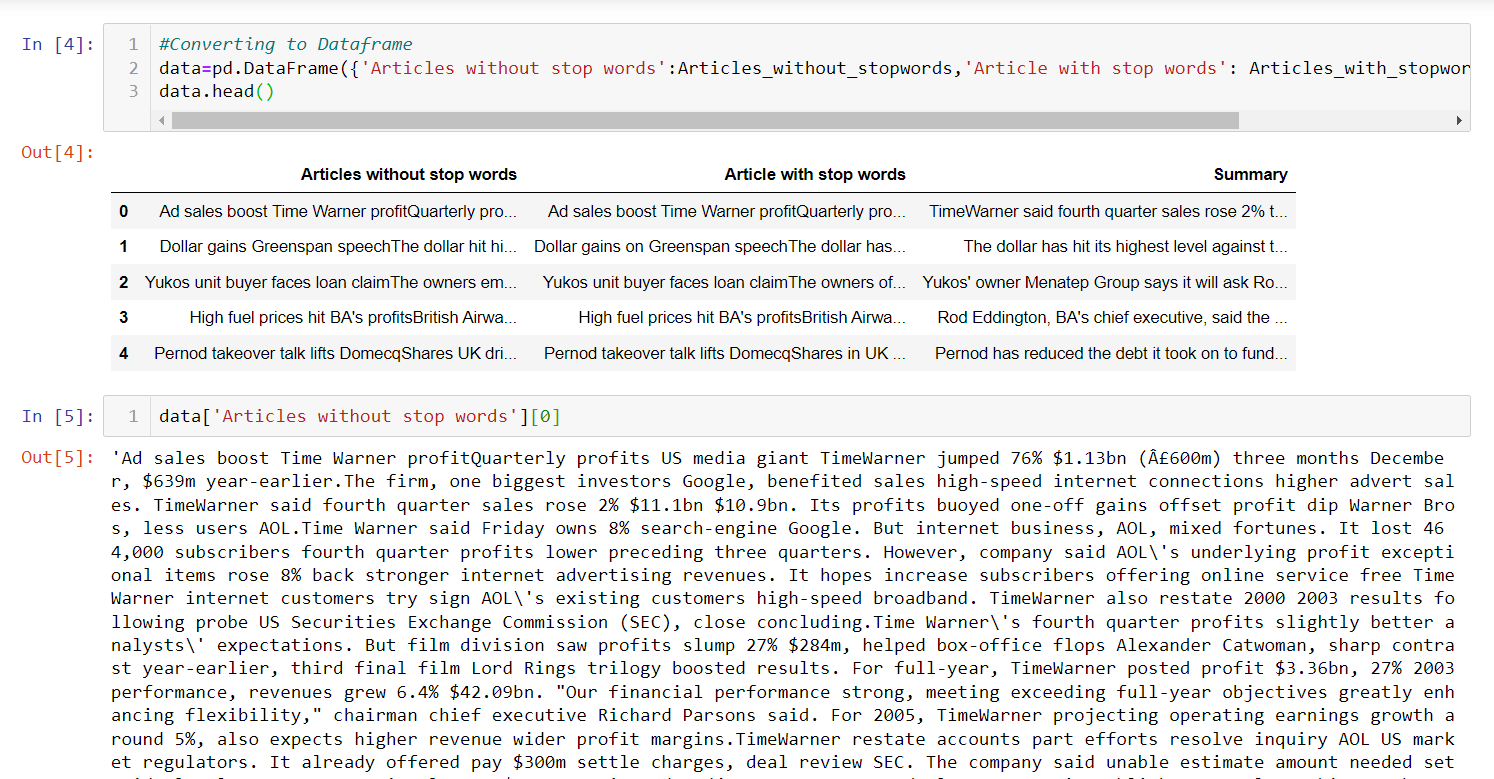
1. **Analyze the dataset**: I investigated the text files from my dataset that is BBC news summary and their respective summaries. The different sectors of news namely business, politics, etc had approx. 400-500 text files. This was a large dataset and hence a powerful algorithm is needed for test summarisation purpose. I studied the existing methods and read research papers.
2. **Loading the Libraries and Data:** The libraries used were numpy, pandas, NLTK, matplotlib, TensorFlow, sklearn, OS, glob, TQDM, etc. I loaded all the text files using the OS, TQDM and glob libraries and created a function to read the articles text files and summaries text file separately from the path of the folder.
3. **Text cleaning and pre-processing**: Prior to moving on to the model development phase, it is crucial to complete some fundamental pre-processing processes. Using sloppy and filthy text data might be a very bad idea. As a result, we will remove all extraneous symbols, letters, etc. from the text in this stage that have no bearing on the problem's main goal. In this step I removed stop words from the text files of articles. Converting them to Dataframe and displaying text files and their respective summaries using range function. I also used contraction mapping and cleaned the text. As Seq2seq model will be used. I added <START> and <END> tokens for the target text that is on the summary files.
4. **Understanding the distribution of the sequences:** Here, I examined the length of the reviews and the summary to obtain a general sense of how long the material is distributed. This will enable to fix the sequence's maximum length. Additionally, I created Histograms to visualize the distribution of the articles and summaries. It was shown that the maximum length of the article was around 500 and the maximum length of summary was 100
5. **Splitting of the dataset**: I split the data into training and testing modules.
6. **Tokenizer**: A tokenizer increases vocabulary while transforming word sequences into numeric sequences. From tensorflow.keras.processing.text I imported tokenizer to create separate tokenisation function for articles text and summaries text.
7. **Building the Model**: Finally, the section where we create the model is here. LSTM generates the hidden state and cell state for each timestep when the return sequences parameter is set to True. State = True on return: When return state is set to true, the LSTM only generates the hidden state and cell state from the previous timestep. Using Initial state, the LSTM's internal states are initialized for the first time. Layered LSTM consists of many LSTM layers stacked on top of one another. This results in a more accurate depiction of the scenario. As the loss function, I am using sparse categorical cross-entropy, which instantly transforms the integer sequence into a one-hot vector. This gets around any memory problems.6 Moreover, I am also monitoring validation loss value.
8. **Set up the Inference for Encoder and Decoder**: After training, new source sequences with unknown target sequences are used to test the model. So, in order to decode a test sequence, we must configure the inference architecture.6
9. **Prediction**: I describe the functions that turn a numeric sequence into a list of words for a generating the summary.
10. **IMPLEMENTATION**



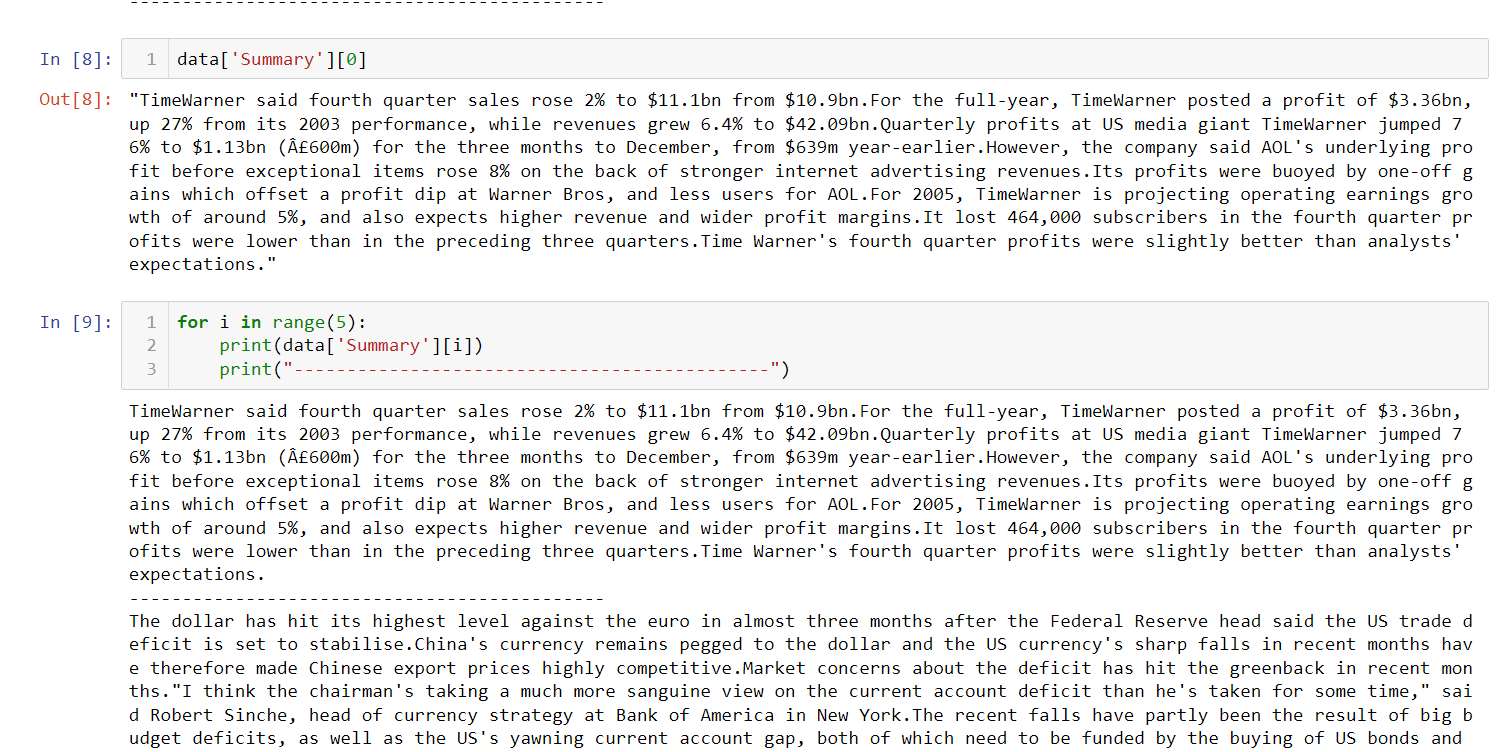
**Importing the libraries**



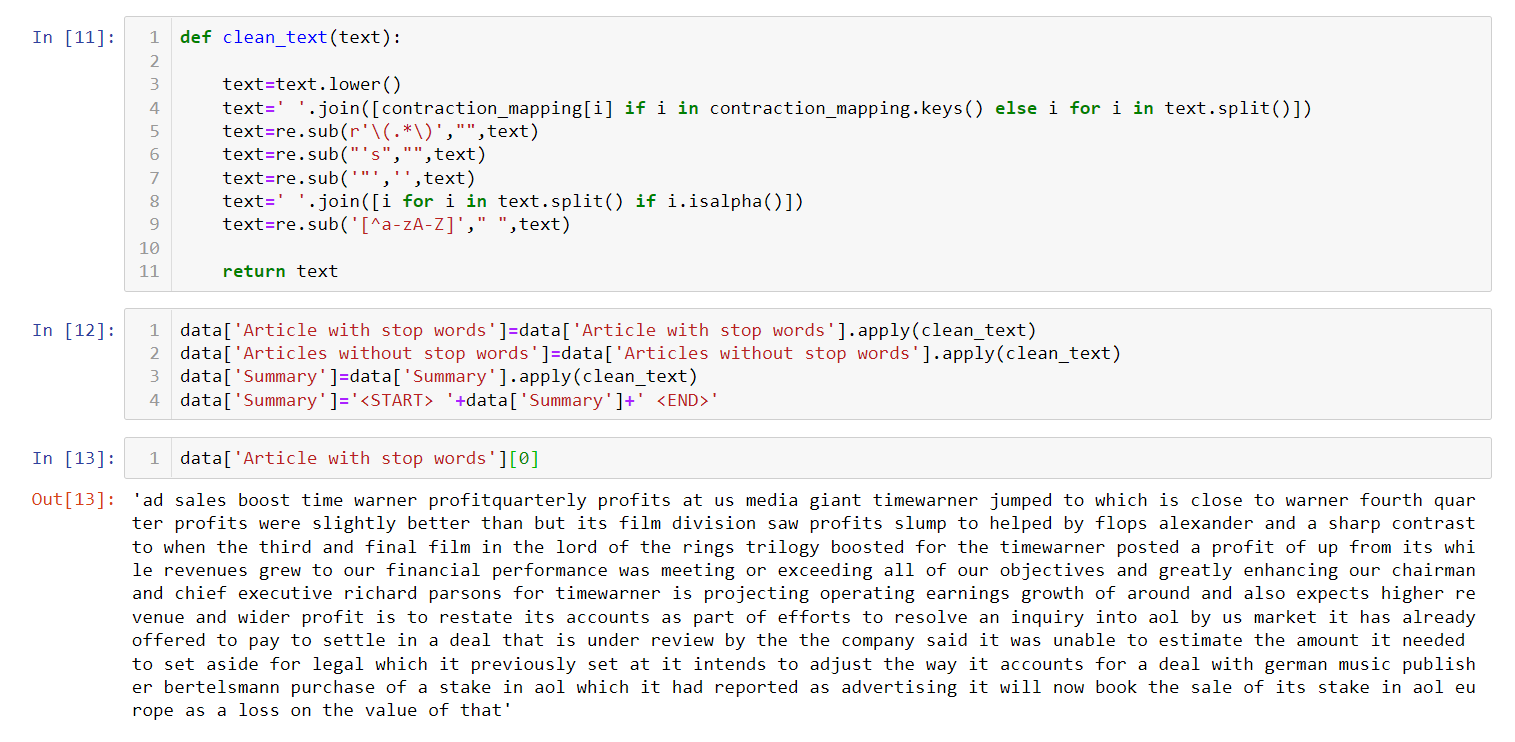
**Loading the dataset**



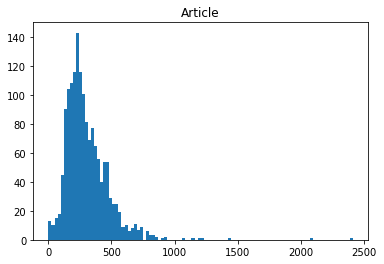
**Displaying the Dataframe and Article text**



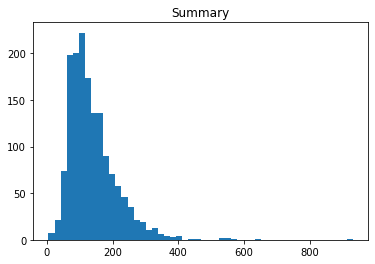
**Showing Summary text**



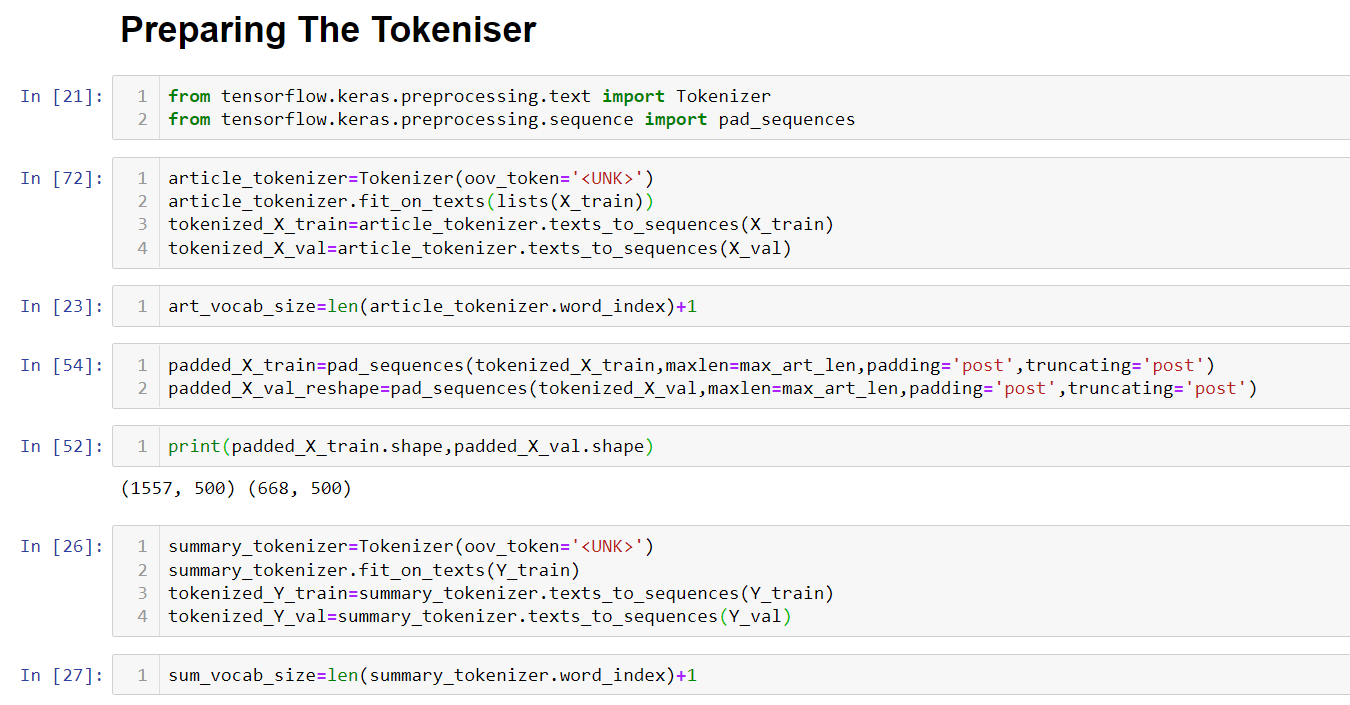
**Cleaning the text**

****

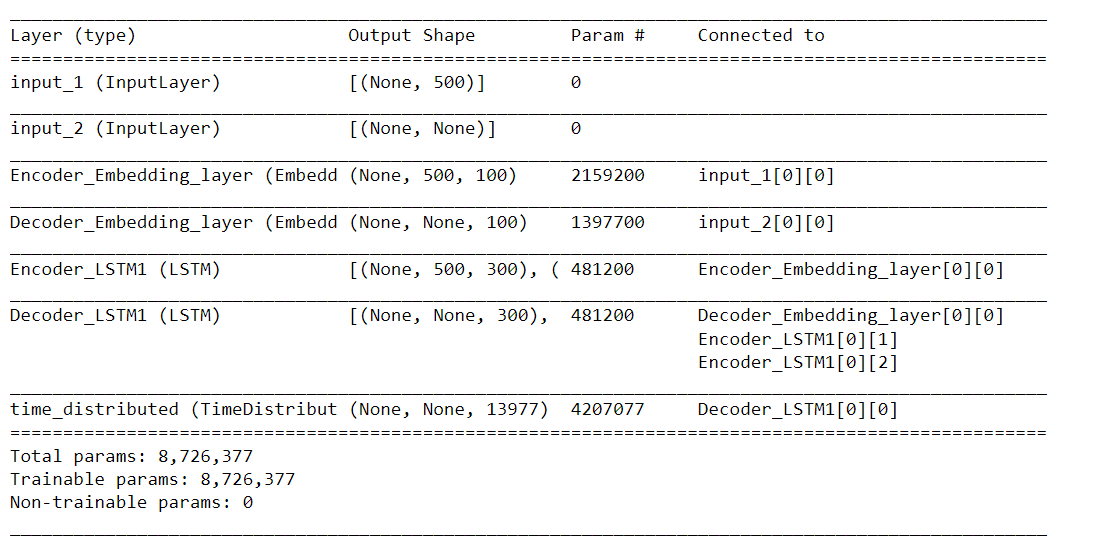
**Length Histogram of Articles text**

****

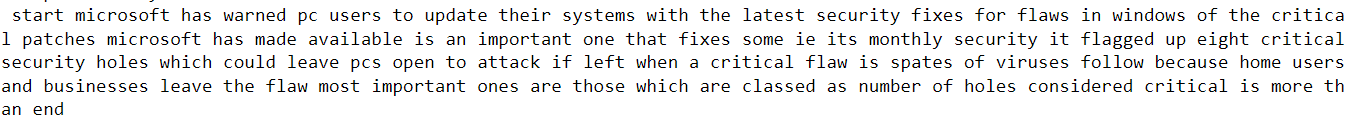
**Length Histogram of Summary**



**Setting the Tokenizer for Article and Summary**



**Setting Encoder and Decoder for the Inference**



**Predicted Summary**

1. **RESULTS**

Initially, I proposed to use TextRank Algorithm for text summarization but after some attempts, for my dataset, in my opinion the algorithm was taking more time and was given multiple errors. Afterwards I discovered Seq2seq model which improved the methodology and showed me better results for the articles w.r.t the summaries. Tokenisation technique works good with text summarization.

1. **CONCLUSION**

For the project definition, my main objective was to convert large amount of text data in the form of articles, to understandable and concise form of summary. I experimented with different context of statements. Information overload is a result of the Internet's popularity and the rapid advancement of technology. If there were powerful text summarizers that produced a user-friendly summary of documents, this issue might be resolved. Therefore, a system that allows a user to quickly find and obtain a summary document is required.

1. **FUTURE SCOPE**

Extractive text summarization is simpler to construct. However, text summarizing using abstractive techniques is more effective since they result in a summary that is difficult to generate but semantically linked. Hence, building a commercial platform for this scenario is a great future research.

1. **REFERENCES**

1. Munot N, S. Govilkar S. Comparative Study of Text Summarization Methods. International Journal of Computer Applications. 2014;102(12):33-37. doi:10.5120/17870-8810

2. Chauhan K. Unsupervised Text Summarization using Sentence Embeddings. Medium. Published January 1, 2019. Accessed December 11, 2022. https://medium.com/jatana/unsupervised-text-summarization-using-sentence-embeddings-adb15ce83db1

3. Pai A. Text Summarization | Text Summarization Using Deep Learning. Analytics Vidhya. Published June 10, 2019. Accessed December 11, 2022. https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarization-using-deep-learning-python/

4. BBC News Summary. BBC News Summary | Kaggle. Accessed December 11, 2022. /datasets/pariza/bbc-news-summary

5. Joshi P. Automatic Text Summarization Using TextRank Algorithm. Analytics Vidhya. Published November 1, 2018. Accessed December 12, 2022. https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/

6. Pai A. Text Summarization | Text Summarization Using Deep Learning. Analytics Vidhya. Published June 10, 2019. Accessed December 12, 2022. https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarization-using-deep-learning-python/