Text Summarization using TF-IDF and Textrank algorithm

Sarika Zaware
Department of Computer Engineering
AISSMS Institute of Information Technology
Pune, India
sarika.zaware@aissmsioit.org

Deep Patadiya
Department of Computer Engineering
AISSMS Institute of Information Technology
Pune, India
deeppatadiya24@gmail.com

Abhishek Gaikwad
Department of Computer Engineering
AISSMS Institute of Information Technology
Pune, India
abhigaikwad001@gmail.com

Sanket Gulhane
Department of Computer Engineering
AISSMS Institute of Information Technology
Pune, India
sankey.gulhane@gmail.com

Akash Thakare
Department of Computer Engineering
AISSMS Institute of Information Technology
Pune, India
akashthakare1999@gmail.com

2021 5th International Conference on Trends in Electronics and Informatics (ICOEI) | 978-1-6654-1571-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICOEI51242.2021.9453071

***Abstract*— In this digital era, a tremendous amount of information gets generated every day. The generated information is used for multiple purposes such as scientific/medical research, news generation, blogs, etc. Reading and gathering the important information and summarizing it manually can become a tedious task which is also prone to manual errors. Therefore, more time is required to read that information, and also unwanted information gets mixed up with important information. It is difficult for a person to manually summarize a large document. Also, there is an issue with finding necessary documents and absorbing relevant information from them. In this proposed system, we are implementing a combination of TFIDF and Textrank algorithm with some NLP methods which will efficiently summarize the given data and will perform better than the other systems.**

Keywords*— *TF-IDF, Text summarization, Sentence ranking, Cosine similarity matrix, Textrank algorithm, Word frequency.

I. Introduction

With the rapidly growing technology, data is getting more valuable every day. Today, the data is gathered and circulated in huge amount. The amount of digital data created yearly is predicted to increase rapidly. With a large amount of information circling in the computerized space, there is a necessity to generate text summarization methods that can shorten longer sentences efficiently [21]. Text summarization is the process of identifying the most important information in a source text and compressing them into a shorter version without changing the significance of the original text. Using text summarization improves reading time, makes information searching process easier, and reduces irrelevant data.

Currently there are two summarization techniques available: Abstractive and Extractive. In abstractive text summarization we build an intermediate semantic representation of the document. Summary is synthesized from the intermediate semantic representation. Summary may not be having original content and may use paraphrasing techniques. The extractive text summarization technique includes picking key phrases from the original document and merging them to create a summary whereas, abstractive text summarization creates its own new phrases and create summary relevant to the source text. We are going to use the Extractive summarization technique. Extractive methods generate a summary using the text which is already the part of the source document(s). It can be accomplished by applying different techniques such as deep learning, fuzzy logic, Neural Networks (NN), C-means clustering, etc. Various methods have been proposed for the summarization task. In this proposed system, we are implementing a mixture of TF-IDF [Term Frequency and Inverse Document Frequency], Textrank algorithm and some NLP [Natural Language Processing] methods which will provide more accurate results than previously implemented text summarization models.

Hence, we are trying to implement an algorithm using combination of two different algorithms to yield more efficient outcomes as compared to other algorithms.

II. Literature Review

J.N.Madhuri et al. [1] has proposed a system which uses sentence ranking to summarize a single document and store the summary in the audio form. It first tokenizes and removes the stop words and that adds parts-of speech tag to every token and every token is given weight and maximum weight and weight frequency of the document is calculated. Then sentences weight is also calculated and is ranked from high to low in descending order. The sentences with high ranks are used for summarization and stored in audio format. It is using weights for calculation purposes which are less complex. The ranking of sentences by summing the weights can generate an abrupt summary because weights of some irrelevant sentences might be more than the relevant sentence. The proposed systemis simple and easy to understand as the calculations are less complex but can generate abrupt summary due to ranking of the sentences.

Shohreh et al. [2] are explaining the relationship between text mining and text summarization and understanding prominent conditions and stages required for good summarization and also evaluating several summarization methods and selecting best out of them. This study gives an idea of important parameters and also methods that can be used for summarization.

Rahul et al. [3] are evaluating various summarization methods which have structure and sematic based approach for text summarization and are also using various datasets with single and multiple documents. They have also discussed common methods used for text summarization like Machine leaning (ml), reinforcement learning, NNs, fuzzy logic, sequence to sequence modeling, etc. and also checking how their accuracy scores are different. The have also discussed about optimization algorithms. They have also mention about how multiple methods work better than single method. The study gives an idea about common methods used for text summarization and how their accuracy scores are also different on same dataset. It also discusses about optimization algorithms and how the multiple methods work better than single method.

Rahul et al. [4] are discussing how various methods are used for text summarization and which are better and why. They have stated about extractive and abstractive summary algorithms and Machine learning methods perform better than any other algorithm. The study shows that why text summarization is important and are comparing different methods are evaluating which is better and why. They also mentioned about the quality of the summary and how different work differently in summarization.

Apra et al. [5] has proposed a system that has the capability of storing, retrieving, and maintaining the information. The word in the document in the collection or corpus is evaluated for its importance by the system. The information retrieval is done using TF-IDF by calculating the TF and IDF weight values and then by examining the TF-IDF weight. The retrieving and ranking of queries is done according to their relevancy in Retrieval and ranking process. After these processes result gets displayed which increases accuracy. in the word-count part the similarity is directly computed which may be reduce speed for vocabularies that are large. Information retrieval systems can prove to be much better by using the TF-IDF. This algorithm would help in increasing the success of query retrieval systems.

N. S. Shirwandkar et al. [6] has proposed a system in which input is taken in .txt format, i.e. text document. It is then pre-processed for sentence segmentation, tokenization, and stop word and punctuation removal. Then to find the score of the sentences their features are calculated. After that the input is given to Restricted Boltzmann Machine and Fuzzy logic. Two separate summaries are generated through this process on which set of operations is performed. However, the complete dependence of fuzzy logic on human knowledge adds a drawback. The final summary generated using this combined method results in better summary then RBM method alone.

P. Janjanam et al. [7] made discussing about machine learning, recent methods which are evolutionary and based on graph for representation of features to selection of sentences and summary generation. The study is meant to help build effective Natural Language Processing applications. The paper covered topics about representation and features of text, graph based summarization, optimization based summarization and rouge scores comparison by various summarized text methods.

Yanxia [8] has proposed a system which is improvement over the traditional TF-IDF method by introducing the coefficient of weight for part of speech tag and the weight of position for the characteristic word. This is done by using the TF-IDF-NL algorithm which has the function for extraction of the characteristic word that proves improvement in retrieval. This algorithm is better, in the effect of clustering which can better reflect the text characteristic. It assumes that independent similarity evidence is provided by the counts of different words. The system effectively improves the effect of clustering of the characteristic words and the textual characteristics are reflected better and can be very beneficial.

Fadi et al. [9] has proposed a system in which different weighting methods are devised to improve the TFIDF technique, using which more appropriate documents can be fetched in the system. In this system weighting in TFIDF is done by three new techniques Dispersed Words Weight Augmentation, Title Weight Augmentation, First Ranked Words Weight Augmentation because of which more relevant documents are retrieved. The performance of retrieval of information is improved. It does not use semantic similarities between words. As the new weighting techniques are better and have high recall values and the weight of words across a document increments, the retrieval can be more efficient.

G. V. Madhuri Chandu et al. [10] have proposed a system that presents a model which retrieves concise and irredundant answers to various questions or queries about educational institutions. It uses various NLPs based techniques for summarization of text to give relevant results and also hybrid similarity measure and clustering algorithms are used. It involves data collection, data pre-process, tokenization, retrieving sentences relevant to query from the original text. It has two phases: 1. Retrieving sentences relevant to query. 2. Redundant sentences removal. This system helps to summarize the content based on the query specified by the user. This model works efficiently in many cases, but there are some cases where the model is unable to retrieve some of the important sentences. It has a limitation of extracting less important or irrelevant sentences from the website.

C. Xu [11] has proposed an NLP based information retrieval method. The description of the natural language under the multiple conditions is done by semi-supervised learning algorithm, and for every data an undirected graph is constructed and algorithm propagation of label is used for the graph simplification and reduce the complexity of computation in the process of retrieval. This system combines semi-supervised learning method and Text Rank algorithm which can effectively improve the efficiency and accuracy of retrieval.

Arfiani et al. [12] has proposed a system which has a web application for information retrieval. This system is proposed to find back the information needed by the system using efficient weighting techniques and TF-IDF. First, the information retrieval is done then focused is on preprocessing stages and then weighting TF-IDF using the initial article. The basic words are then calculated by using the TF-IDF weighting method to find out the weight value of each article. The occurrence frequency of words in a document provided shows importance of the word in the document. The system returns several documents and all the information is relevant, but large numbers of other relevant documents are ignored. As all the relevant information and document can be retrieved by using the TF-IDF and its weighting technique this can prove very beneficial.

Animesh Ramesh et al. [13] proposes a combination of graph and intersection-based algorithm. In order to summarize text it uses statistical and semantic analysis for calculating importance of textual units in large data sets. The methodology includes pre-processing, generation of vectors, generation of semantic graphs, generating scores and at the end obtaining summary. The advantage of this methodology is that it does not rely upon any primitive datasets for assessment. The disadvantage of the Sentence Rank is that the time for iterations through the synsets is more. The paper follows a better approach for graph calculation as it develops semantic diagrams by utilizing implied links which depend upon the semantic relatedness between text nodes.

Nasser Alsaedi el at. [14] proposes techniques for microblog documents summarization by selecting the posts that are most relevant. In this methodology a temporal TF-IDF is calculated that produces a summary without the need of earlier knowledge of the entire dataset. The proposed temporal TF-IDF technique performs better than all the other summarization systems for both. This method can be very efficient, as the temporal method achieves good results in ROUGE evaluation scores and in human evaluation scores.

Lu Yao el at. [15] has proposed a keyword extraction system takes English news text as input. It joins TF-IDF and the Textrank for keyword extraction and summary calculation. This involves TF-IDF for generating vectors and rankings. Text rank is used to create the graph-based model and generate a summary by considering the weight of the sentences. The combination of these algorithms can effectively improve the efficiency of keyword extraction. Textrank and TF-IDF combination can yield better results than normal Textrank. It makes extraction of keywords easier and helps in determining weights efficiently.

K Usha Manjari el at. [16] proposes a method which takes user-based query details which is extracted from different websites over the internet for generating an extractive summary. Firstly, the query data is extracted through selenium, then by using pre-processing and TF-IDF algorithm the summary is generated. This methodology is distinctive and efficient for obtaining summaries according to request of user, as selenium is used for web scraping. This method summarizes the data of multiple web pages related to a user particular query, hence obtaining a quick result of the topic searched.

S. Jabri et al. [17] proposes a method that utilizes vector space model for extracting the relevant information. It operates by weighting TF-IDF for allotting a score to a document and to rank the records. It has two phases. The initial phase is association rules generation and the second one is text document ranking. In English text, to convert words into their grammatical root form porter stemmer algorithm is used. For ranking text document, combination of TF-IDF measure in vector space model is used. Combining classical ranking approach and association rules significantly improves the information retrieval system performance. The optimal time is not promised by the generation of association rules, but the time is still less than the execution time of the semantic approach ESA (Explicit Semantic Analysis). This proposed system uses association rules generation and ranked documents based on vector space model.

Y. Wang et al. [18] proposes an improved way for TF-IDF algorithm, based on domain knowledge graph. This proposed system will use the legal knowledge graph to obtain better results for TF-IDF algorithm, so that perfect weight assigned can be known to the particular domain-related keywords. This system uses TF-IDF which is a commonly used algorithm, knowledge graph which is based on the data set associated with it, and other knowledge for base support. Through in-depth semantic analysis and mining with association of data resources, it establishes a network of relationships. Major advantage of this system is that the combination of TF-IDF algorithm and improved domain knowledge graph method which helps in significantly improving the domain-related vocabulary weight as compared to other improved methods which leads to improvement of accuracy.

F. Yamout et al. [19] has proposed a system in which three new weighting techniques DWWA, TWA and FRWWA are involved to refine the TFIDF weighting technique. The DWWA technique is used to increment weight of words spread across a document, TWA in words document adds more weight to words and FRWW to check most frequent words in every document. Major advantage of this proposed system is that it generates better result than the point of reference at higher recall values. This system is used for improving the precision than the baseline at higher recall values as it is relevant for search engines which provides more suitable documents on the first pages.

Jeena Jacob [20] proposes a Cap-Net based gated multi-task learning, to remove the irrelevant attributes. This is said to significantly improve the predictions and the classification accuracy. Caps-Net based Gated-MTL is used for classification of text. The capsule network is used in text classification which uses dynamic routing. It involves layers like primary-capsules and class capsules. The Caps-Net for the text is received in text samples which are embedded. The capsule network with the cluster feature prediction uses the vectors which enhances text classification. The proposed system shows that it provides a better text classification compared to the other methods.

We have studied 20 different research papers for understanding various text summarization algorithms. The proposed algorithms in those papers have one or more limitations which can be improved. So to overcome these limitations, we are proposing TF-IDF using Textrank algorithm to generate better extractive summary.

III. PROPOSED SYSTEM

In this paper we have proposed an automated text summarization system which helps in reducing the unnecessary efforts required to summarize a document or a piece of text. The system architecture is shown in Figure 1.

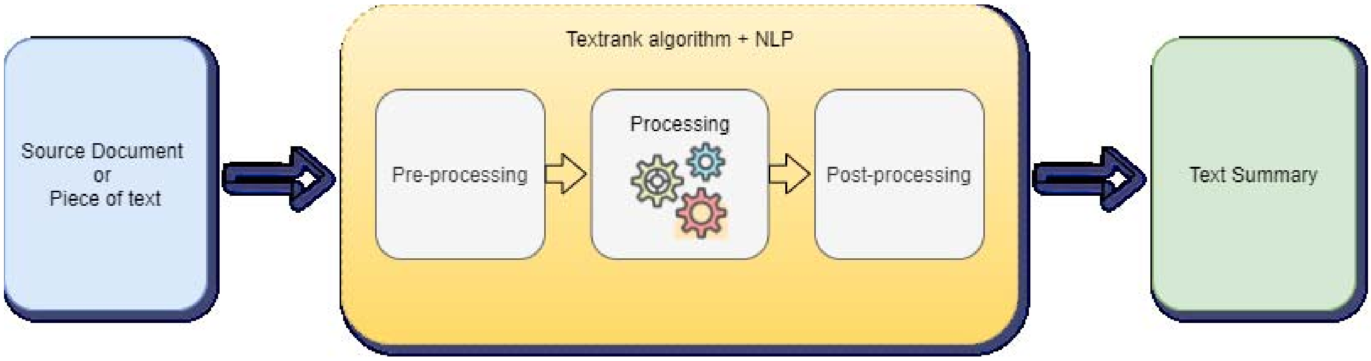


Figure 1. System Architecture

The user will first give the input for summarization. This input can be a text file, a pdf file or a piece of text. After giving the input, the first step will be pre-processing, i.e. preparing the data by cleaning and normalizing the text for further processing. The cleaning step will include tokenizing the sentences, removing stopwords, removing punctuations and extra white spaces. Following this, normalization will be done by lower-casing all the text data and performing stemming or lemmatization. After this the processing stage occurs, i.e. calculating unique words, words per sentences, frequency of the words, TF matrix, IDF matrix and TF-IDF matrix. The next part is post processing, i.e. calculating cosine similarity, graph creation and score generation using Textrank and sentence ranking.

A. Details about the modules

I. Preprocessing:

In this stage, the information which irrelevant for summarization or which can affect the final result is removed.

1. Data cleaning:

- **Tokenization:**
  - **Sentence tokenization:** It means splitting the given text into sentences and storing them in a list.

    Eg.: "My name is Edward. I am a data scientist."

    After sentence tokenization: ['My name is Edward.', 'I am a data scientist.']

  - **Word tokenization:** It means splitting the given text into words and storing them in a list.

    Eg.: "My name is Edward."

    After word tokenization: ['My', 'name', 'is', 'Edward', '.']

- **Removing punctuations and extra white spaces:** Removing punctuations is necessary because it holds no meaning value in the data and can also create and issue in differentiating words. Sometimes white spaces may also create problems in using the data so removing them is a good practice.

  Eg.: "Hi! My name is   Edward."

  After removing punctuations and extra white spaces: Hi My name is Edward

- **Removing shortwords:** The shortwords are words which contain apostrophe in them. So to convert them in into their normal form we perform this step.

  Eg.: "I've been working as data scientist for six years."

  After removing shortwords: I have been working as data scientist for six years.

- **Removing stopwords:** The stopwords are words which are most common in any language. The words like he, she, and but, etc. are called stopwords. The use of these words does not hold any meaning to the data so they must be removed to reduce the feature.

  Eg.: "I am a data scientist."

  After removing stopwords: ['I', 'data', 'scientist', '.']

2. Normalization:

- **Lowercasing all the words:** The lowercasing of data is a necessary step because languages like Python are case sensitive. So 'HELLO' and 'hello' will be considered different.

Eg.: "I am a data scientist."

After lowercasing all the words: i am a data scientist.

- **Lemmatization:** It converts the word to it root form called lemma. This lemmatizing process, brings the word into its dictionary format. This is more accurate and makes analysis better.

  Eg.: studying, studies, etc.

  After lemmatizing the word: study

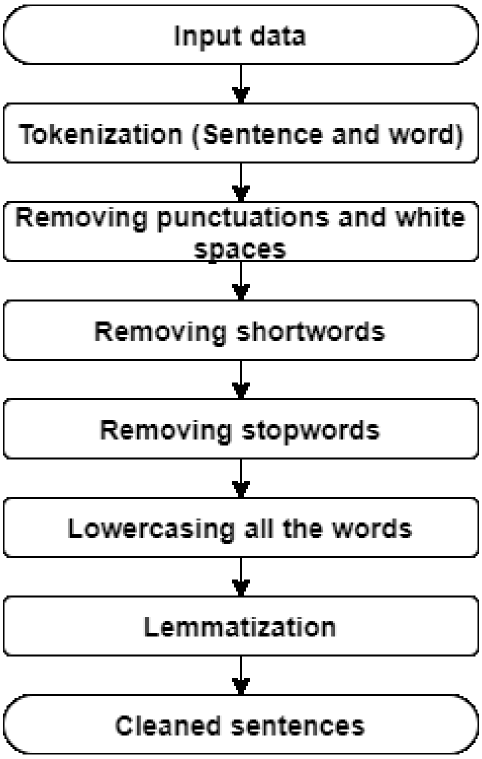

Input data
Tokenization (Sentence and word)
Removing punctuations and white spaces
Removing shortwords
Removing stopwords
Lowercasing all the words
Lemmatization
Cleaned sentences

Figure 2. Flowchart of pre-processing stage

II. Processing

This is next step after processing. In this stage, calculation of unique words, word count of sentences, frequency of words and TFIDF matrix will be calculated.

1. **Unique words calculation:** This step creates a list of all distinctive words present in the given text data after doing all the preprocessing. It is used while calculating TFIDF matrix.
2. **Word count per sentences:** This step creates a dictionary which contains length of all the sentences by calculating the number of words per sentences after all the preprocessing.
3. **Frequency of the words:** This step calculates the frequency of a word in a sentence and stores it in a dictionary format.
4. **Calculating TFIDF matrix:** TFIDF is a numerical statistic that is used to show the importance of a word in the document in a collection or corpus [22].

- **TF matrix:** TF stands for Term Frequency. It calculates how many times the word occurs in the document.

  Term Frequency (TF) = Number of times a term(t) appears in a document / Total number of words in the document (1)

- **IDF matrix:** IDF means Inverse Document Frequency. It computes the prominence of the word in the document.

  Inverse Document Frequency (IDF) = log(Total number of document(N)/Number of documents which has the term(t) in it)
  (2)

- **TFIDF matrix:** TFIDF is the product of Term Frequency(TF) and Inverse Document Frequency(IDF).

  Term Frequency-Inverse Document Frequency (TFIDF) = TF*IDF (3)

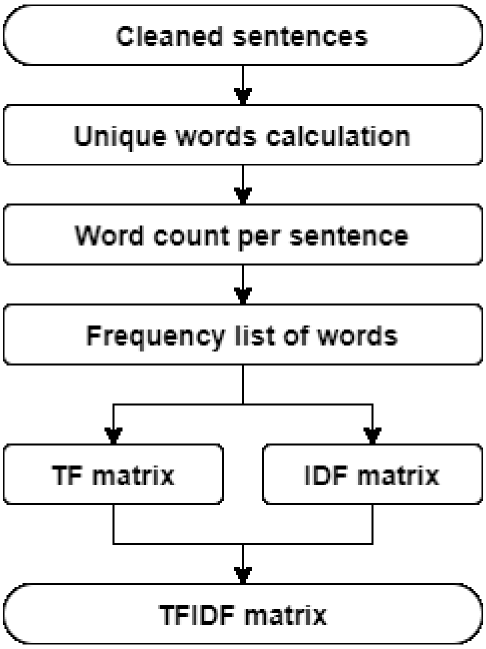


Figure 3. Flowchart of processing stage

III. Post processing

This is the next step after processing. In this step, we calculate the cosine similarity matrix, generate graph, generate scores using Textrank algorithm and rank sentences for summary generation.

1. **Cosine similarity:** Cosine similarity is required in NLP to measure the text-similarity between two documents without considering their size. A word is represented into a vector form and the text documents

are represented in n-dimensional vector space. The Cosine similarity of two documents will range from 0 to 1 [23].

The range of similarity is from 0 to 1 (0 is least similar and 1 is most similar).

Cosine similarity = cos(θ) =

$$\frac{A \cdot B}{\|A\| \, \|B\|} \tag{1}$$

2. **Graph creation:** The graph is created based on the cosine similarity matrix and is passed to pagerank function for score generation. In the graph, the similarity scores are edges and sentences are vertices, for ranking sentences. For graph generation, "networkx" library is used. The graph in Figure 4., is of one of files from the input dataset.

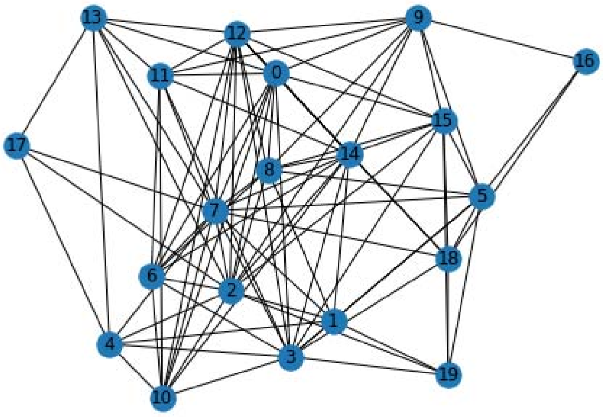


Figure 4. Graph generated for one of the files from the input dataset

3. **Score generation using Textrank algorithm and Sentences Ranking:** Textrank algorithm is a graph-based ranking model for text processing which can be utilized to find the most important sentences in text and furthermore to discover important keywords [24].

It is based on PageRank algorithm. The pagerank function from "networkx" library is used to generate scores. The generated scores are generated in dictionary format.

After the scores are generated, the scores are sorted in descending order and their key value is taken as index to select the sentences from list of tokenized sentences.

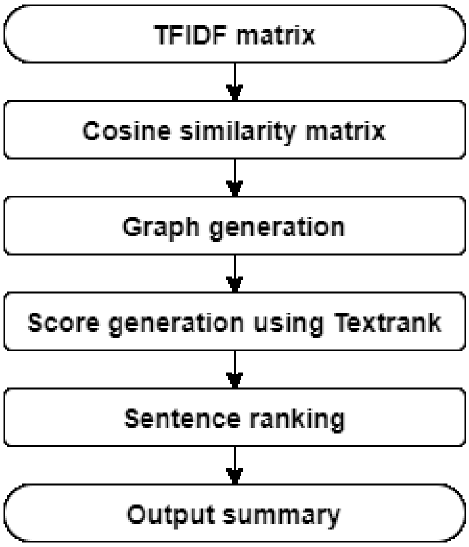


Figure 5. Flowchart of post-processing stage

B. Algorithm

These are the steps required to implement the proposed algorithm. The steps are as follows:

1. The user gives a text input for summarization.
2. Tokenize the input in sentence token and word token
   i. *sentences_list = sent_tokenize(input_data)*
   ii. *word_list = word_tokenzie(input_data)*
3. For every sentence in sentence_list:
   i. *cleaned_data.append(re.sub(r'[^\w\s]', '', sentence)), i.e. remove punctuations*
   ii. *cleaned_data.append(sentence.replace('-', ' ')), i.e. remove extra whitespaces.*
4. After this remove stopwords.
5. For every sentence in cleaned_data:
   i. *For word in sentence: if stopwords then continue, else store in cleaned_data.*
6. For every sentence in cleaned_data:
   i. *For word in sentence: word.lower(), lemma.lemmatise(word)*
7. After this we create unique words list by checking unique words in the input list.
8. Now, we create a dictionary of words per sentence.
   i. *details = {"doc_id" : ind, "doc_length" : count}*
9. After this we construct a dictionary of frequency of the words in the input data.
10. The next step includes calculating TFIDF matrix.
    i. *TF matrix*
    ii. *IDF matrix*
    iii. *TFIDF matrix*
11. Now we calculate cosine similarity matrix using TFIDF matrix as input.
12. After this we generate a graph which will be used for sentence score generation.
    i. *nx_graph = nx.from_numpy_array(cosine_similarity_matrix)*
13. Using the graph, as an input for Textrank algorithm the dictionary of score of sentences is generated.

i. *scores = nx.pagerank(nx_graph)*

14. Then, scores of dictionary are used to rank the sentences.
15. After these stages, a 7-line output summary will be generated.

This is how the summary is generated by the proposed algorithm.

C. Dataset

The dataset used for summarization purpose is BBC new summary dataset from Kaggle. The dataset contains 4450 files in which 2225 files are original text and remaining 2225 files are summaries of the original text. From this data set we are selecting the first 10 files as sample input for evaluation and comparison of the proposed system and TF-IDF algorithm.

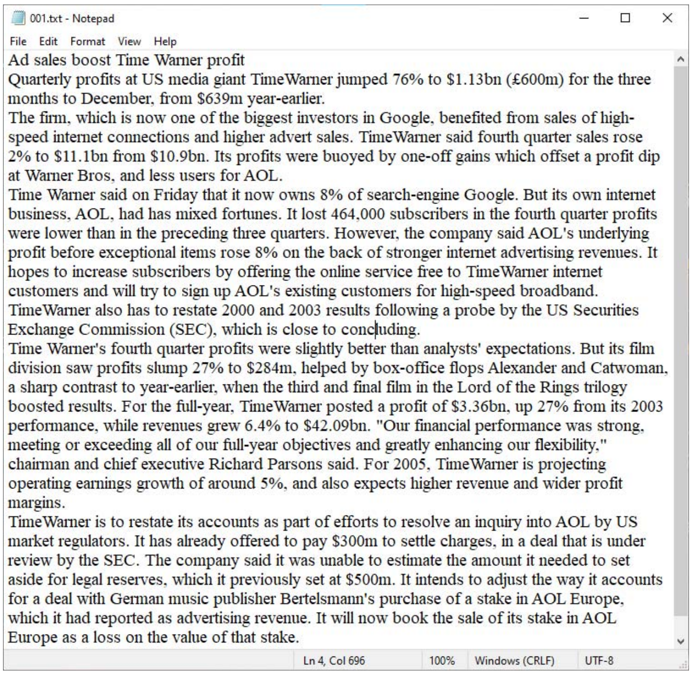
IV. RESULT AND DISCUSSION

The result of the implemented system hence provides multiple benefits and more efficient summary generation than available algorithms. We have evaluated the proposed algorithm with TFIDF algorithm using Rouge-Score evaluation.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation), is an algorithm which is used for evaluating text summarization and machine translated output in NLP. It compares the candidate summary against a reference summary for calculating the score of the text summarization [25]. The following equation shows the Rouge-N formula:

$$\text{Rouge-N} = \frac{\sum_{C \in R_S} \sum_{n-gram \in C} Count_{match}(n-grams)}{\sum_{C \in R_S} \sum_{n-gram \in C} Count\ (n-grams)} \tag{1}$$

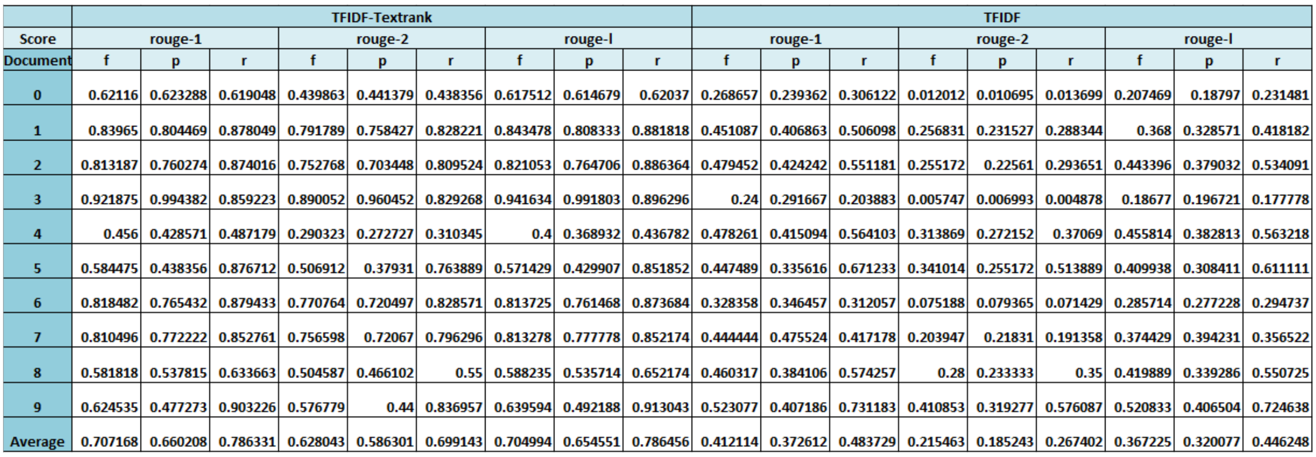
Where C stands for candidate summary, R_S stands for Reference summaries, n in n-gram stands for length, Count(n-grams) in number of co-occurrences in candidate and reference summary. For calculating result of proposed algorithm (TFIDF-Textrank algorithm) and TFIDF algorithm, we are giving 10 sample paragraphs as input from BBC dataset. The Figure 6., shows one of the input text data file from the sample dataset.



Ad sales boost Time Warner profit
Quarterly profits at US media giant TimeWarner jumped 76% to $1.13bn (£600m) for the three months to December, from $639m year-earlier.
The firm, which is now one of the biggest investors in Google, benefited from sales of high-speed internet connections and higher advert sales. TimeWarner said fourth quarter sales rose 2% to $11.1bn from $10.9bn. Its profits were buoyed by one-off gains which offset a profit dip at Warner Bros, and less users for AOL.
Time Warner said on Friday that it now owns 8% of search-engine Google. But its own internet business, AOL, had has mixed fortunes. It lost 464,000 subscribers in the fourth quarter profits were lower than in the preceding three quarters. However, the company said AOL's underlying profit before exceptional items rose 8% on the back of stronger internet advertising revenues. It hopes to increase subscribers by offering the online service free to TimeWarner internet customers and will try to sign up AOL's existing customers for high-speed broadband. TimeWarner also has to restate 2000 and 2003 results following a probe by the US Securities Exchange Commission (SEC), which is close to concluding.
Time Warner's fourth quarter profits were slightly better than analysts' expectations. But its film division saw profits slump 27% to $284m, helped by box-office flops Alexander and Catwoman, a sharp contrast to year-earlier, when the third and final film in the Lord of the Rings trilogy boosted results. For the full-year, TimeWarner posted a profit of $3.36bn, up 27% from its 2003 performance, while revenues grew 6.4% to $42.09bn. "Our financial performance was strong, meeting or exceeding all of our full-year objectives and greatly enhancing our flexibility," chairman and chief executive Richard Parsons said. For 2005, TimeWarner is projecting operating earnings growth of around 5%, and also expects higher revenue and wider profit margins.
TimeWarner is to restate its accounts as part of efforts to resolve an inquiry into AOL by US market regulators. It has already offered to pay $300m to settle charges, in a deal that is under review by the SEC. The company said it was unable to estimate the amount it needed to set aside for legal reserves, which it previously set at $500m. It intends to adjust the way it accounts for a deal with German music publisher Bertelsmann's purchase of a stake in AOL Europe, which it had reported as advertising revenue. It will now book the sale of its stake in AOL Europe as a loss on the value of that stake.

Figure 6. One of input file from sample dataset

This dataset contains original text paragraphs and their sample summarizes. The generated summary is a 7-line output. These are the following scores generated using rouge score evaluation.



	TFIDF-Textrank									TFIDF								
Score	rouge-1			rouge-2			rouge-l			rouge-1			rouge-2			rouge-l		
Document	f	p	r	f	p	r	f	p	r	f	p	r	f	p	r	f	p	r
0	0.62116	0.623288	0.619048	0.439863	0.441379	0.438356	0.617512	0.614679	0.62037	0.268657	0.239362	0.306122	0.012012	0.010695	0.013699	0.207469	0.18797	0.231481
1	0.83965	0.804469	0.878049	0.791789	0.758427	0.828221	0.843478	0.808333	0.881818	0.451087	0.406863	0.506098	0.256831	0.231527	0.288344	0.368	0.328571	0.418182
2	0.813187	0.760274	0.874016	0.752768	0.703448	0.809524	0.821053	0.764706	0.886364	0.479452	0.424242	0.551181	0.255172	0.22561	0.293651	0.443396	0.379032	0.534091
3	0.921875	0.994382	0.859223	0.890052	0.960452	0.829268	0.941634	0.991803	0.896296	0.24	0.291667	0.203883	0.005747	0.006993	0.004878	0.18677	0.196721	0.177778
4	0.456	0.428571	0.487179	0.290323	0.272727	0.310345	0.4	0.368932	0.436782	0.478261	0.415094	0.564103	0.313869	0.272152	0.37069	0.455814	0.382813	0.563218
5	0.584475	0.438356	0.876712	0.506912	0.37931	0.763889	0.571429	0.429907	0.851852	0.447489	0.335616	0.671233	0.341014	0.255172	0.513889	0.409938	0.308411	0.611111
6	0.818482	0.765432	0.879433	0.770764	0.720497	0.828571	0.813725	0.761468	0.873684	0.328358	0.346457	0.312057	0.075188	0.079365	0.071429	0.285714	0.277228	0.294737
7	0.810496	0.772222	0.852761	0.756598	0.72067	0.796296	0.813278	0.777778	0.852174	0.444444	0.475524	0.417178	0.203947	0.21831	0.191358	0.374429	0.394231	0.356522
8	0.581818	0.537815	0.633663	0.504587	0.466102	0.55	0.588235	0.535714	0.652174	0.460317	0.384106	0.574257	0.28	0.233333	0.35	0.419889	0.339286	0.550725
9	0.624535	0.477273	0.903226	0.576779	0.44	0.836957	0.639594	0.492188	0.913043	0.523077	0.407186	0.731183	0.410853	0.319277	0.576087	0.520833	0.406504	0.724638
Average	0.707168	0.660208	0.786331	0.628043	0.586301	0.699143	0.704994	0.654551	0.786456	0.412114	0.372612	0.483729	0.215463	0.185243	0.267402	0.367225	0.320077	0.446248

Figure 7. Performance comparison between TFIDF-Textrank algorithm and TFIDF algorithm

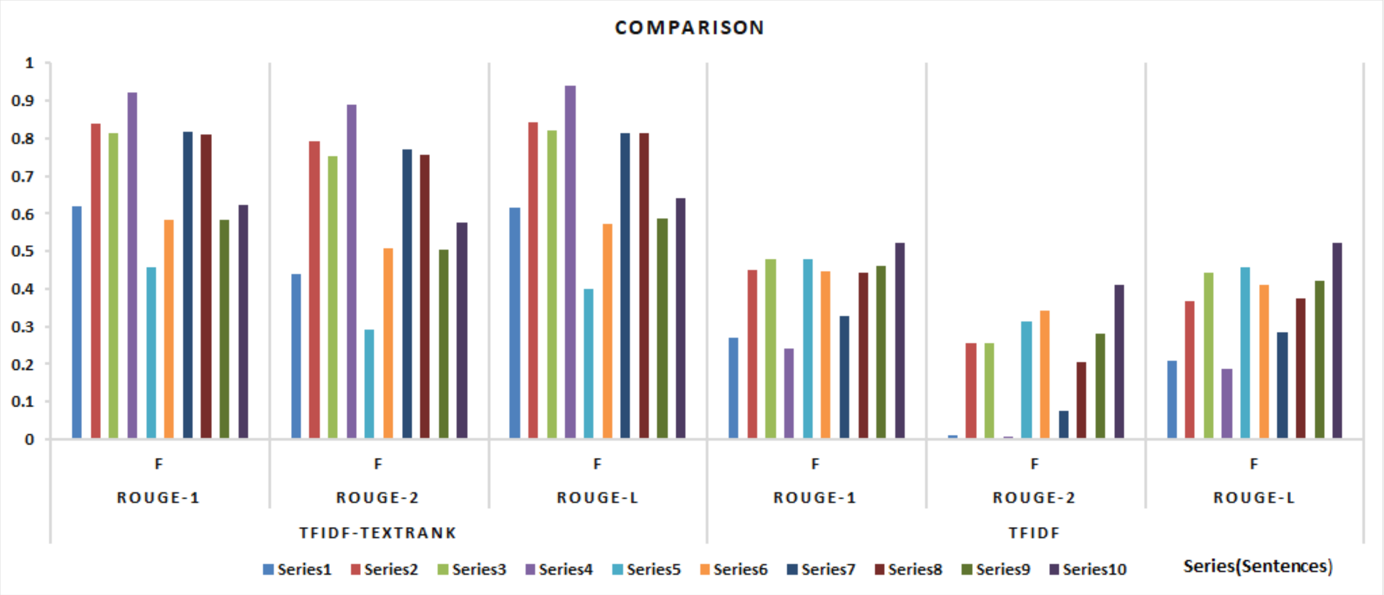


Figure 8. Comparison of result between TFIDF-Textrank algorithm and TFIDF algorithm

As shown in Figure 7., TFIDF-Textrank algorithm is performing better than TFIDF algorithm. The average scores are also good of TFIDF-Textrank algorithm as compared to TFIDF algorithm. The result comparison of proposed algorithm and TFIDF algorithm is also done in Figure 8.

V. CONCLUSION

The implemented system provides a cost effective solution for generating extractive summary using TF-IDF, text rank algorithm and some NLP methods. It saves reading time, helps in gathering of relevant information, and improves the task of researching for information. The text summarization can be further used in applications such as media monitoring, newsletters, patent research etc., We have successfully evaluated and compared the proposed system using rouge score evaluation technique. The results of our system were giving better efficiency than TF-IDF algorithm. The algorithms currently available for text summarization are not yet perfect and still lack the expected summary generation. The proposed system can also be further developed for improvement in rouge score. Currently it can be used in applications such as media monitoring, customer support bots, e-learning, etc.

References

[1] J. N. Madhuri and R. Ganesh Kumar, "Extractive Text Summarization Using Sentence Ranking," 2019 International Conference on Data Science and Communication (IconDSC), Bangalore, India, 2019, pp. 1-3, doi: 10.1109/IconDSC.2019.8817040.

[2] S. R. Rahimi, A. T. Mozhdehi and M. Abdolahi, "An overview on extractive text summarization," 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, Iran, 2017, pp. 0054-0062, doi: 10.1109/KBEI.2017.8324874

[3] Rahul, S. Adhikari and Monika, "NLP based Machine Learning Approaches for Text Summarization," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2020, pp. 535-538, doi: 10.1109/ICCMC48092.2020.ICCMC-00099.

[4] Rahul, S. Rauniyar and Monika, "A Survey on Deep Learning based Various Methods Analysis of Text Summarization," 2020 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2020, pp. 113-116, doi: 10.1109/ICICT48043.2020.9112474.

[5] A. Mishra and S. Vishwakarma, "Analysis of TF-IDF Model and its Variant for Document Retrieval," 2015 International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, India, 2015, pp. 772-776, doi: 10.1109/CICN.2015.157.

[6] N. S. Shirwandkar and S. Kulkarni, "Extractive Text Summarization Using Deep Learning," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-5, doi: 10.1109/ICCUBEA.2018.8697465.

[7] P. Janjanam and C. P. Reddy, "Text Summarization: An Essential Study," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 2019, pp. 1-6, doi: 10.1109/ICCIDS.2019.8862030.

[8] Y. Yang, "Research and Realization of Internet Public Opinion Analysis Based on Improved TF - IDF Algorithm," 2017 16th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES), Anyang, China, 2017, pp. 80-83, doi: 10.1109/DCABES.2017.24.

[9] F. Yamout and R. Lakkis, "Improved TFIDF weighting techniques in document Retrieval," 2018 Thirteenth International Conference on Digital Information Management (ICDIM), Berlin, Germany, 2018, pp. 69-73, doi: 10.1109/ICDIM.2018.8847156.

[10]] G. V. Madhuri Chandu, A. Premkumar, S. S. K and N. Sampath, "Extractive Approach For Query Based Text Summarization," 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, India, 2019, pp. 1-5, doi: 10.1109/ICICT46931.2019.8977708.

[11] C. Xu, "Research on Information Retrieval Algorithm Based on TextRank," 2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Jinzhou, China, 2019, pp. 180-183, doi: 10.1109/YAC.2019.8787615.

[12] A. N. Khusna and I. Agustina, "Implementation of Information Retrieval Using Tf-Idf Weighting Method On Detik.Com's Website," 2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA), Yogyakarta, Indonesia, 2018, pp. 1-4, doi: 10.1109/TSSA.2018.8708744.

[13] A. Ramesh, K. G. Srinivasa and N. Pramod, "SentenceRank — A graph based approach to summarize text," The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014), 2014, pp. 177-182, doi: 10.1109/ICADIWT.2014.6814680.

[14] N. Alsaedi, P. Burnap and O. Rana, "Temporal TF-IDF: A High Performance Approach for Event Summarization in Twitter," 2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI), 2016, pp. 515-521, doi: 10.1109/WI.2016.0087.

[15] L. Yao, Z. Pengzhou and Z. Chi, "Research on News Keyword Extraction Technology Based on TF-IDF and TextRank," 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), 2019, pp. 452-455, doi: 10.1109/ICIS46139.2019.8940293.

[16] K. U. Manjari, S. Rousha, D. Sumanth and J. Sirisha Devi, "Extractive Text Summarization from Web pages using Selenium and TF-IDF algorithm," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020, pp. 648-652, doi: 10.1109/ICOEI48184.2020.9142938.

[17] S. Jabri, A. Dahbi, T. Gadi and A. Bassir, "Ranking of text documents using TF-IDF weighting and association rules mining," 2018 4th International Conference on Optimization and Applications (ICOA), 2018, pp. 1-6, doi: 10.1109/ICOA.2018.8370597.

[18] Y. Wang, D. Zhang, Y. Yuan, Q. Liu and Y. Yang, "Improvement of TF-IDF Algorithm Based on Knowledge Graph," 2018 IEEE 16th International Conference on Software Engineering Research, Management and Applications (SERA), 2018, pp. 19-24, doi: 10.1109/SERA.2018.8477196.

[19] F. Yamout and R. Lakkis, "Improved TFIDF weighting techniques in document Retrieval," 2018 Thirteenth International Conference on Digital Information Management (ICDIM), 2018, pp. 69-73, doi: 10.1109/ICDIM.2018.8847156.

[20] Jacob, I. Jeena. "Performance Evaluation of Caps-Net Based Multitask Learning Architecture for Text Classification" Journal of Artificial Intelligence 2, no. 01 (2020): 1-10.

[21] https://towardsdatascience.com/a-quick-introduction-to-text-summarization-in-machine-learning-3d27ccf18a9f [24/3/2021]

[22] https://en.wikipedia.org/wiki/Tf%E2%80%93idf [24/3/2021]

[23] https://studymachinelearning.com/cosine-similarity-text-similarity-metric/ [24/3/2021]

[24] https://cran.r-project.org/web/packages/textrank/vignettes/textrank.html [24/3/2021]

[25] https://en.wikipedia.org/wiki/ROUGE_(metric) [24/3/2021]