International Journal of Grid and Distributed Computing
Vol. 14, No. 1, (2021), pp. 1524–1538

News Article Summarization: Analysis and Experiments on Basic Extractive Algorithms

*Tameem Ahmad[1], Sayyed Usman Ahmed[2], Nesar Ahmad[3], Areeba Aziz[4], Lakshita Mukul[5]

Department of Computer Engineering, Z. H. College of Engineering & Technology, Faculty of Engineering and Technology, Aligarh Muslim University, Aligarh, 202002, India

[1]*tameemahmad@gmail.com,* [2]*syedusmanahmed@zhcet.ac.in,* [3]*nesar.ahmad@gmail.com,* [4]*areebaa@tejasnetworks.com,* [5]*lakshitamukul197@gmail.com,*

Abstract

A web is an information system that stores an enormous number of documents and other online resources. We generally access these documents and resources by Uniform Resource Locators (URLs) over the Internet. There was a time when people used to wait for newspapers in order to catch the previous day's happenings but thanks to the internet, all the latest information is now available with a click of a button. However, the information is much larger than one can manage quickly and efficiently. Also today everyone wants to gain more information in less time. Instead of reading large documents and then getting the insight, it is better to read a summary that gives the core information about the topic and helps in gaining more information in less time. In this paper we have implemented three techniques for generating the extractive summary of news articles of the two benchmark datasets- CNN-corpus and the BBC (these datasets have both the article and its summary) on different retention rates in order to know what would be the best retention rate for generating the summary which would contain most of the information of the original text without much affecting the connectivity among the sentences since, readability and connectivity are the two prime factors because of which most of the people still rely on man written summaries.

Keywords: *Extractive Text Summarization, Information extraction, Lexical Chains, Natural Language Processing, News Summarization, TextRank, TF-IDF*

1. Introduction

As of June 2020 [1], massive web pages are indexed on the World Wide Web and approximately contain at least 5.47 billion pages. Further massive online data is generating each day with immense velocity, volume and variety [24]. This enormous data creation requires deepened techniques for accessing information from this vast amount of resources i.e. improved techniques for Information Retrieval (IR). With the ease in availability of Internet and increased use of smart devices like mobile phone, laptop, tablets the access and storage of data has escalated rapidly, resulting in need of tools which can enhance the user's productivity and experience [16].

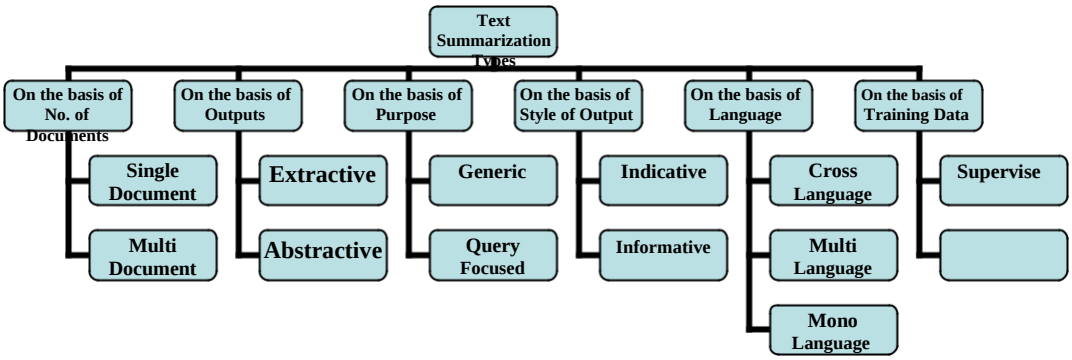
Text summarization is the technique to produce shorter replacements for a longer text that are usually hard to read manually for extracting useful information from large text [19]. Summarization has been identified as an effective method [15, 17], which helps users to locate the right information at the right time, thus facilitating timely decisions.

Automatic text summarization techniques can be used for extracting keywords. Extracting keywords using a text summarization algorithm can optimize the summarization process [19]. However, text summarization methods have their own pros and cons. We have used three techniques namely TextRank, Lexical Chain and TF-IDF in this work.

ISSN: 2005-4262IJGDC

2. Existing Work

Automatic Text Summarization is the process of generating a gist of a document automatically by the system itself without compromising with the meaning of the original document [1]. An automated generated summary can be of various types depending on the purpose and kind of data available for summarization. This categorization of text summarization techniques is depicted in Fig. 1 below [2].



ISSN: 2005-4262IJGDC

Figure 1. Summarization Types

The basic classification of text summarization techniques based on its nature of summary generation i.e. it can either be of abstractive or extractive nature. In the Abstractive Technique of text summarization it is supposed to understand the concept of the document by the system and then generate a new short text for summary of that original document that must be precise and in simple natural language [14,18]. For this technique, the newly generated text of the summary doesn't include the same sentences from the original document/text.

On the other hand, extractive text summarization does not generate new sentences rather it uses the most important sentences from the original document exactly and discards the rest. Deciding important sentences plays key role in the efficiency of the summarization i.e. how good is the generated summary is that can absolutely represent the original document precisely. The sentences from the original text are ranked on the basis of its linguistics also the statistics plays important role in it. A lot of research is found on the extractive type of summarization [3].

2.1. Summarization Methods

Due to the inherent complexity of generating abstractive summaries, extractive summaries have been more frequently generated and used in practical applications [2]. Various types of methods can be used to generate extractive summaries such as statistical-based methods, graph-based methods [25], discourse-based methods, topic-based methods, machine learning-based methods and swarm-based or optimization-based techniques [3].

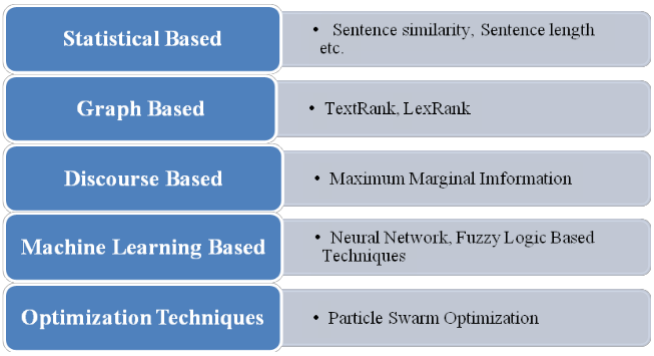


Figure 2. Summarization Methods

Summarization methods have been briefly explained below.

- Statistical Based Methods: The methods generate summaries using statistical features of the document like sentence position, centrality of the sentence, sentence length, numeric data in a sentence, title similarity etc. [1]. These techniques do not require much storage or fast processors and that language-independent.

- Graph-Based Methods: In this method, a graph of (v, e) is created where the vertices represent words or sentences and the edges between them represents the similarity relationship between these nodes. The sentences to be taken in extractive summary are found by traversing the graph and selecting the sentences which have a similarity index above the defined threshold. Some of the recognized graph-based methods are TextRank and LexRank [4].

- Discourse Based Methods: These methods require understanding the textual structure. It becomes complex to use as they consider the sentence connections and their parts within a document [1].

- Topic-Based Methods: In this method, the summary is generated by firstly identifying the subject or theme of the document. Then this is used to extract the sentences which are related to the subject [1].

ISSN: 2005-4262IJGDC

- Machine Learning-Based Methods: These include approaches where the machine learns to produce the summary from data provided to it. These data can either be supervised where the training data is provided with the summary and the machine can learn how to produce the summary from this training data or unsupervised where the machine learns by analyzing the document since no training data is available. Some of the machine learning techniques are neural network, SVM, Genetic algorithm and fuzzy logic [4].

- Optimization techniques: These techniques use nature-inspired algorithms such as swarm algorithms and are usually used in combination with other techniques [3].

3. AUTOMATIC TEXT SUMMARIZATION TECHNIQUES

This section describes the techniques used for automatic text summarization as follows:

3.1. Lexical Chain

A lexical chain technique forms a group of related words that represents the concept of the document. This group formation is logical and based on semantic relations due to synonyms, or hypernyms/hyponyms [6, 20, 23]. For example, the two words kept together could be due to one of the following reason:

- Two words are exactly the same along with same grammatical sense.
- Two noun instances are not the same but their meaning is same (i.e. are synonyms).
- The two instances have the same hypernym/hyponym relation between them. A hypernym is defined as a generalized word representing a category where more specific words may fall inside. (For example, animal and cat)
- The two noun word instances are related as siblings in the hypernym/hyponym tree. (cat and dog) [6]

3.2.TF-IDF

TF-IDF depicts the importance of a word in a given document of collection. It is a mathematical calculation for scoring the importance of a word based on its frequency of appearance in the document to that of the collection of documents.

Formula for calculating TF and IDF:

$$TF(w) = \frac{No.of\ times\ term\ w\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document} \quad (1)$$

$$IDF = \log_{10}\left(\frac{Total\ No.of\ documents}{No.of\ documents\ with\ term\ w\ in\ it}\right) \quad (2)$$

Hence TF-IDF for a word can be calculated as:

$$TF - IDF(w) = TF(w) * IDF(w) \quad (3)$$

3.3. TextRank

TextRank algorithm is influenced by PageRank algorithm. A graph creation takes place using natural language processing. This graph represents the relationship between the entities of the text. A vertex's importance is calculated based on the count of the number of edges to it from another vertex. i.e. the score calculation for a vertex is by counting the inbounds and outbounds edges of that vertex. Mathematically, the ranking for the vertices, that are association score, are calculated by the given formula [5]:

$$S(V_i) = (1-d) + d * \sum_{j \in \ln(v_i)} \left(\frac{1}{|Out(V_j)|}\right) S(V_j) \quad (4)$$

Where, G = (V, E) represents a directed graph

V = set of Vertices

E = set of Edges

ISSN: 2005-4262IJGDC

In (Vi) = inbound of vertex Vi

Out (Vi) = outbound of vertex Vi

d ∈ [0, 1] is a damping factor in which d is a probability the node visits a neighboring node and (1 − d) is the probability of jumping from one vertex to some random vertex. It is generally assigned as 0.85 [5].

However, we cannot measure the strength of the connection between two vertices. Thus, a new formula is introduced which incorporates the weight of the edges while computing the score of vertexes and is given as:

$$WS(Vi) = (1-d) + d * \frac{wij}{\sum_{V_k \in Out(V_j)} W_{jk}} * WS(Vj) \quad (5)$$

where, ( ) - weighted score for vertex .

- weight, which represents the strength of the connection between two vertices and .

4. PROPOSED FRAMEWORK

In this section, we have illustrated the approaches used in this research work, which include an overview of the research undertaken, an architectural view of this work, and details about each module and its algorithm.

4.1. Proposed Framework

In this work, we have implemented three different algorithms i.e., Lexical Chains, TF-IDF and TextRank and have generated the summaries on different retention rates i.e., 30-percent, 40-percent and 50-percent on two different benchmark datasets namely BBC news dataset and CNN news dataset. All three algorithms have been run on all the articles present in both datasets for three different retention rates.

From each dataset, we picked up the articles one by one and has performed some pre-processing on them. After preprocessing, we extract the keywords which give the central idea of the article and based on these keywords; we score the individual sentence and then we have selected the top n sentences to form the summary. The value of n is dependent on the retention rate percentage.

4.2. Dataset

To begin the work, there are specific preparatory steps taken for implementing the proposed model. First and foremost was the data collection for which we required a moderately large amount of meaningful textual content. In this work, we were assessing the performance of three algorithms on newspaper articles. The two datasets that we selected are the CNN news dataset and the BBC news dataset. Both the datasets were stored on Google Drive.

4.2.1. BBC News Dataset

We have used a public dataset of news articles from the BBC from 2004 to 2005 comprised of 2225 articles; each article is also labeled in one of the five predefined categories that are: business, entertainment, politics, sport or tech. For each article, a golden summary is provided in the Summaries folder for reference and evaluation purposes.

4.2.2. CNN News Dataset

The CNN corpus developed by Lins and colleagues [8, 9] contains extensive global news articles. Currently, this corpus consists of more than 2000 texts arranged into eleven categories: Africa, Asia, business, Europe, Latin America, Middle East, US, sports, tech, travel, and world news. These texts are the news article from the CNN website (http://www.cnn.com). The beauty of this corpus is its high quality, conciseness, general interest, up-to-date subject, clarity, and linguistic correctness. Additionally, a good quality summary is also provided with each text

ISSN: 2005-4262IJGDC

which is called highlights. These highlights are short as three to four sentences. These highlights can also serve as a reference for the evaluation and comparison purpose [2, 10].

The reasons for the selection of these datasets were the variety of non-related articles present in the dataset and the hand-written summaries of these articles provided with the dataset.

4.3. Architectural View

The proposed approach firstly retrieves articles from both of our datasets and then preprocesses them by performing tokenization and removing the unwanted words. For Keyword extraction, the three algorithm processes the document in their own way. Finally, according to the retention rate, top n sentences are selected for generating the summary. Fig. 5 shows an overview of the overall system.

ISSN: 2005-4262IJGDC

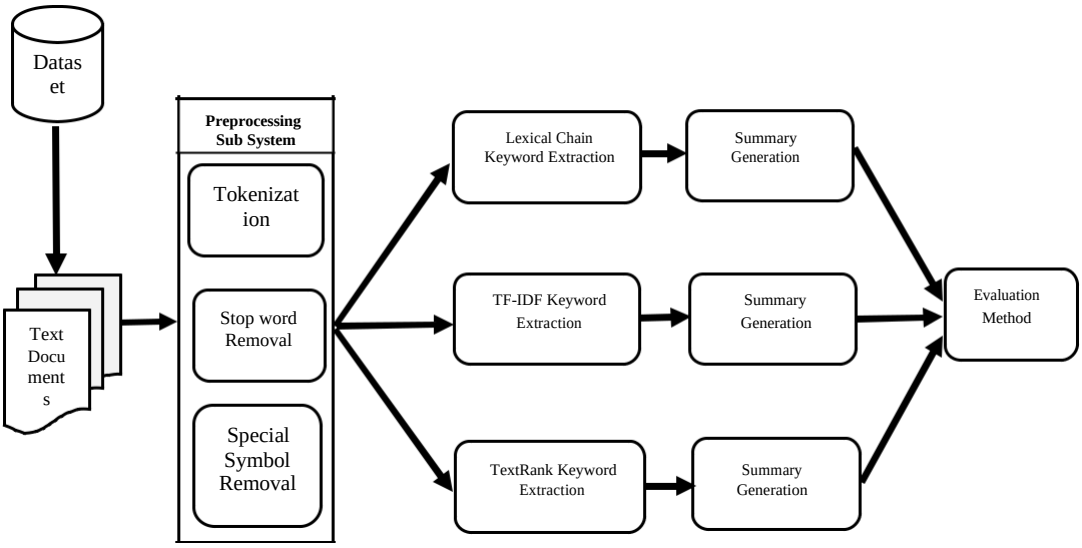


Figure 3. Architectural View

4.4. Evaluation Method

We have used ROUGE for evaluation as it is mostly used for quantitative evaluation of the summaries generated by different scoring methods.

The following is the five evaluation metrics:

- ROUGE-N: is based on the N-gram model with co-occurrence statistics.
- ROUGE-L: It stands for the Longest Common Subsequence (LCS) based statistics. It considers structure similarity at the sentence level and figures out the longest co-occurring in n-gram sequence.
- ROUGE-W: It is a statistical LCS with a weight that considers successive LCSes.
- ROUGE-S: It is a statistical Skip-bigram method that considers co-occurrence. A pair of words in their sentence order is a skip-bigram.
- ROUGE-SU: Skip-bigram along with statistics of unigram-based co-occurrence.

For our experiments, we use the ROUGE-N metric, with N=1, since we are only concerned with single words extracted. Formally, we can say that ROUGE-N is an n-gram recall model among a possible applicant summary to the group of reference summaries [1]. The general formula for calculating the metric is:

$$ROUGE - N = \frac{\sum_{s \in ref} \sum_{gram\ n \in s} Count_{match}(gram_n)}{\sum_{s \in ref} \sum_{gram\ n \in s} Count(gram_n)} \quad (6)$$

ISSN: 2005-4262IJGDC

N: Length of n-gram

Count: Maximum N-grams in the summary

5. IMPLEMENTATION

In this section, we will discuss the experimental setup of the project done. The first part will discuss the datasets. In the next part, the implementation procedure is discussed, followed by tools used for coding.

5.1. Algorithms

All the three algorithms (Lexical Chain, TD-IDF and TextRank) were implemented independently for three different retention rates and the summaries generated by these algorithms [22] were written in an output file that too was stored on Google drive and these summary files along with the golden summaries were passed into ROUGE for evaluation purpose.

5.1.1. Lexical Chain:

Lexical Chain takes an input and generates a lexical chain after pre-processing, the input in our case was a new article. After tokenization, tagging, and noun filtering, we get a lexical chain as shown in Fig. 6.

```
Chain 1 : {'interests': 2}
Chain 2 : {'lot': 2}
Chain 3 : {'locker': 1, 'room': 1}
Chain 4 : {'courts': 2, 'court': 1}
Chain 5 : {'job': 1, 'jobs': 1}
Chain 6 : {'everyone': 3}
Chain 7 : {'tour': 3}
Chain 8 : {'players': 1, 'player': 1}
Chain 9 : {'tennis': 5}
Chain 10 : {'friends': 5, 'person': 1, 'match': 1, 'girl': 1, 'players': 1, 'women': 1, 'player': 1}
```

Figure 4. Lexical Chain

To make the summary, we used the lexical chain. We have computed the frequency of all the words in the text, taking into account that if a word is in a chain, we count the sum of the whole chain [11, 12, 13]. Once all the frequencies have been computed, we have normalized the values and performed filtering with a threshold both for maximum and minimum.

As a final result, we had put in order the n most important sentences of our text that contained the largest amount of lexical chain and most frequent words in our text. In our case, value of n is based on the retention rate percentage that we have taken [26,27,28] i.e., we have picked the top n most important sentences to generate the output summary.

The number of sentences were calculated as follows:

$$n = \left(\frac{a * b}{100}\right) \quad (7)$$

a: percentage of retention rate i.e., 30,40 or 50
b: total no. of sentences presents in the article

ISSN: 2005-4262IJGDC

n: total no. of sentences in the final summary

Fig. 7 shows the final summary of sentences.

```
Percentage of information to retain(in percent):40
No of sentences in final summary :  6
```

Figure 5. Number of sentences in final summary

Given below is the generated summary for the mentioned sample input article on 40-percent retention rate using lexical chains.

```
When I'm on the courts or when I'm on the court playing, I'm a competitor
and I want to beat every single person whether they're in the locker room
or across the net.So I'm not the one to strike up a conversation about the
weather and know that in the next few minutes I have to go and try to win
a tennis match.Uhm, I'm not really friendly or close to many players.I
have not a lot of friends away from the courts.'When she said she is not
really close to a lot of players, is that something strategic that she is
doing?I think just because you're in the same sport doesn't mean that you
have to be friends with everyone just because you're categorized, you're a
tennis player, so you're going to get along with tennis players.I think
everyone just thinks because we're tennis players we should be the
greatest of friends.
```

Figure 6. Summary based on Lexical Chain

5.1.2. TF-IDF

In TF-IDF approach we the score of the sentence on the basis of TF-IDF score of the words. Based on this score we select the most import sentences which are then merged to form a summary. Below are the TF, IDF, and TF-IDF scores for our news article.

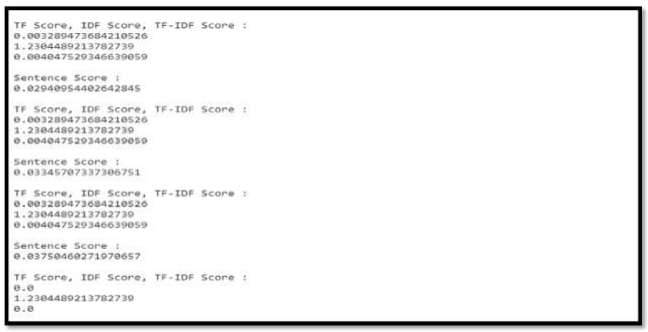


Figure 7. TF, IDF, and TF-IDF scores

Using the TD-IDF, we found the top n sentences as follows:

ISSN: 2005-4262IJGDC

```
sentence NO., sentence score
((12, 0.15752869496446234), (4, 0.07469042200183664), (10, 0.07014482458221077), (3,
0.0502706248171804), (2, 0.045192505699294), (5, 0.04394460433493830:
```

Figure 8. Top n sentences with their scores

Given below is the generated summary using TF-IDF for the mentioned sample input article on 40-percent retention rate using TF-IDF

```
The Russian player has no problems in openly speaking about it and in a
recent interview she said: 'I don't really hide any feelings too much.
I think everyone knows this is my job here.
When I'm on the courts or when I'm on the court playing, I'm a competitor
and I want to beat every single person whether they're in the locker room
or across the net.So I'm not the one to strike up a conversation about the
weather and know that in the next few minutes I have to go and try to win
a tennis match.
I'm a pretty competitive girl.
Is it different on the men's tour than the women's tour?
I think just because you're in the same sport doesn't mean that you have
to be friends with everyone just because you're categorized, you're a
tennis player, so you're going to get along with tennis players.
```

Figure 9. Generated Summary based on TF-IDF

5.1.3. TextRank:

In this algorithm, the first step is to break the complete text into individual sentences. Then, a vector representation is made corresponding to each sentence by word embedding [21]. The next step is the similarity computation between these sentence vectors and then store them in a similarity matrix. This similarity matrix can be transformed into a graph. The vertices of this graph will represent the sentences and edges represent the similarity scores. Finally, some high-ranked sentences, say n, will form the summary [7].

```
Sentence Vector :
[array([ 5.14825583e-02,  1.10544682e-01,  6.94999397e-01,  1.89168096e-01,
        -9.58077684e-02,  3.20288986e-01,  2.70662010e-01,  5.42440832e-01,
        -3.05938005e-01, -1.56364068e-01,  3.70127618e-01,  8.09492469e-02,
         8.41393881e-03,  2.47571543e-01, -3.69342804e-01, -7.61044994e-02,
         8.08582604e-02,  2.30643645e-01, -2.70402402e-01,  5.13828397e-01,
        -6.12548441e-02,  3.87900352e-01,  1.03121363e-01,  7.72494674e-01,
         2.59960234e-01, -7.96069205e-02,  1.42143592e-01, -9.62644577e-01,
         7.54904330e-01,  6.03260659e-02, -4.58570123e-01,  2.36780301e-01,
         2.29152635e-01, -1.56453326e-01,  3.97632688e-01, -2.32720934e-02,
        -5.05520999e-01,  4.13252831e-01, -2.85759270e-01, -1.35231465e-01,
        -1.37098104e-01, -1.48972601e-01,  3.37537557e-01, -3.49540442e-01,
         1.53484434e-01, -2.33341649e-01, -1.98460802e-01, -1.27821520e-01,
         5.08063912e-01, -3.68636876e-01, -2.28472307e-01, -3.15306723e-01,
         1.36149466e-01,  2.22507194e-01,  1.19500056e-01, -1.71007359e+00,
        -1.04403630e-01,  3.45346779e-01,  5.54419458e-01,  7.91236103e-01,
        -2.63593912e-01,  5.01183808e-01, -1.54918820e-01,  2.39762396e-01,
        -4.94388118e-02, -1.39404953e-01, -6.96142530e-03,  4.52243447e-01,
         1.44498184e-01, -1.88242078e-01,  1.62694290e-01,  2.51032356e-02,
        -1.29925504e-01, -2.16523811e-01, -1.39851749e-01,  1.79708660e-01,
```

Figure 10. Sentence Vector

```
Similiarity Matrix :
[[0.         0.64378339 0.59156996 0.7475453  0.63747269 0.61883837
  0.69372612 0.68542236 0.64739478 0.79788285 0.         0.83178782
  0.55843717 0.58564365 0.79771841 0.7565254  0.575041   ]
 [0.64378339 0.         0.83267683 0.85698557 0.71764094 0.81240582
  0.8420344  0.81021047 0.90473139 0.71734643 0.         0.8244583
  0.85043436 0.85658038 0.82539409 0.78656042 0.84938735]
 [0.59156996 0.83267683 0.         0.82313865 0.75527066 0.73470169
  0.79021472 0.80083597 0.89334065 0.60797113 0.         0.84927762
  0.8517375  0.75710064 0.89685303 0.69299048 0.84113187]
 [0.7475453  0.85698557 0.82313865 0.         0.76459354 0.82157695
  0.86001402 0.87651867 0.88054091 0.78653705 0.         0.91478753
  0.86048663 0.82472861 0.86071944 0.85267258 0.77808994]
 [0.63747269 0.71764094 0.75527066 0.76459354 0.         0.68546808
  0.74824065 0.72055054 0.76297432 0.66111451 0.         0.80672944
  0.75617725 0.71717036 0.77864099 0.72995365 0.71190387]
 [0.61883837 0.81240582 0.73470169 0.82157695 0.68546808 0.
  0.84673887 0.81307095 0.84076226 0.70893741 0.         0.82324845
  0.77039582 0.80494732 0.7927357  0.74317771 0.82921433]
 [0.69372612 0.8420344  0.79021472 0.86001402 0.74824065 0.84673887
  0.         0.83118278 0.8988784  0.73612404 0.         0.88512784
  0.8125782  0.83526862 0.86516029 0.79789072 0.84964389]
 [0.68542236 0.81021047 0.80083597 0.87651867 0.72055054 0.81307095
  0.83118278 0.         0.84124988 0.71694529 0.         0.87815398
  0.82030129 0.85208088 0.85451305 0.81738484 0.84847867]
```

ISSN: 2005-4262IJGDC

Figure 11. Similarity Matrix having

As discussed above, the similarity matrix is converted into a graph where graph nodes are the representation of the sentences from the text and the edges are depicting similarity scores among the sentences. Further, PageRank algorithm is used for ranking the sentences. Finally, selected the top n sentences on the basis of their rankings for the sake of summary generation.

{0: 0.05418868552905123, 1: 0.06291967979381004, 2: 0.06107074622671161, 3: 0.06462055979515273, 4: 0.057678951524171324, 5: 0.06075695308614737, 6: 0.06364813003

Figure 12. Sentence score after applying PageRank algorithm

Ranked sentence:
[(0.06606217237750724, "I think just because you're in the same sport doesn't mean that you have to be friends with everyone just because you are categorized, you

Figure 13. Sentence Ranking

When I'm on the court or when I'm on the court playing, I'm a competitor and I want to beat every single person whether they're in the locker room or across the net." So I'm not the one to strike up a conversation about the weather and know that in the next few minutes I have to go try to win a tennis match. Uhm, I'm not really friendly or close to many players. Is it different on the men's tour than the women's tour? I think just because you're in the same sport doesn't mean that you have to be friend with everyone just because you're categorized, you're a tennis player, so you're going to get along with tennis players. I think everyone just thinks because we're tennis players we should be the greatest of friends.

Figure 14. Summary based on TextRank

6. Results

The algorithms mentioned above were run multiple times for three different retention rates for a better understanding of the results on both datasets. Table 1 and 2 summarizes the average results of the aforesaid algorithms on the two datasets.

ISSN: 2005-4262IJGDC

Table 1. Average Results of Algorithm on BBC News Dataset

Algorithm	Average Recall	Average Precision	Average F-measure
Lexical Chain	0.694157237	0.681280743	0.67729453
TF-IDF	0.46554668	0.546465313	0.491573077
TextRank	0.658863517	0.683789443	0.659882693

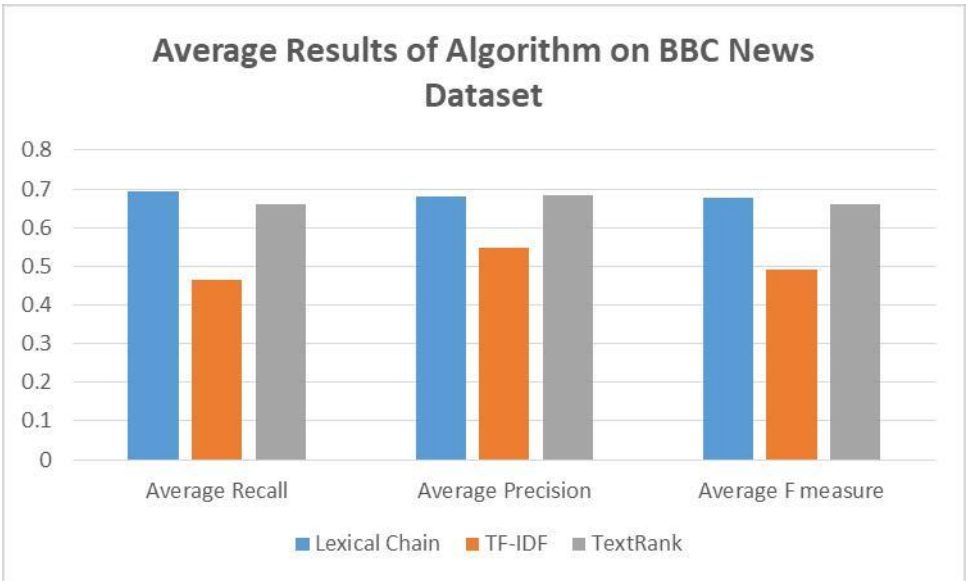
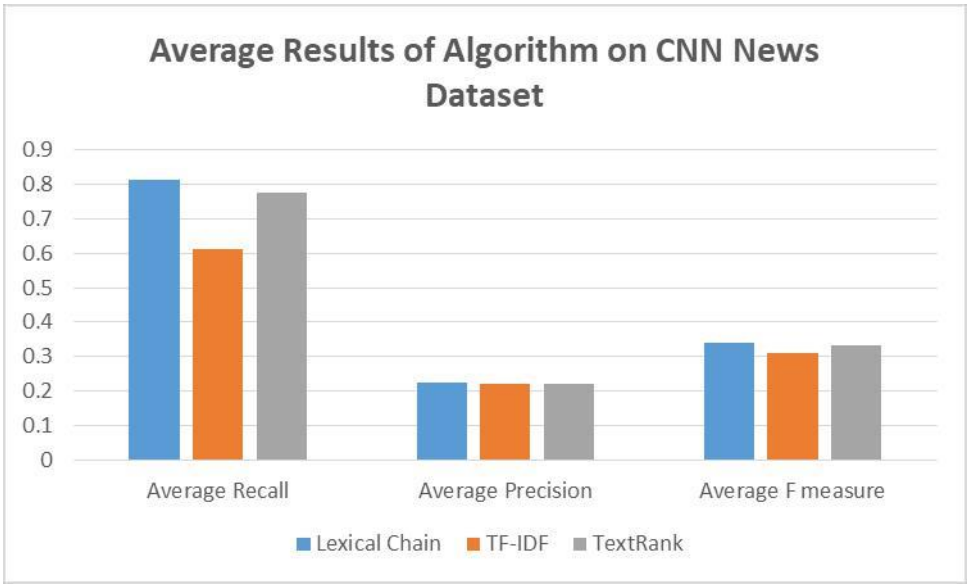


Figure 15. Average Results Depiction in Bar Chart of Algorithms on BBC News Dataset

From our experiment, the Lexical Chain got the highest values for Average Recall, Average Precision and Average F-measure on BBC News Dataset.

Table 1. Average Results of Algorithm on CNN News Dataset

Algorithm	Average Recall	Average Precision	Average F-measure
Lexical Chain	0.81256928	0.223642972	0.338666297
TF-IDF	0.612789693	0.221225553	0.310238801
TextRank	0.777226147	0.221300148	0.33167814



ISSN: 2005-4262IJGDC

Figure 16. Average Results Depiction in Bar Chart of Algorithms on CNN News Dataset

From our experiment, the Lexical Chain got the highest values for Average Recall, Average Precision and Average F-measure on CNN News Dataset.

7. CONCLUSION AND FUTURE WORK

We conclude that this work gives the details of extractive text summarization for newspaper articles using three different algorithms i.e., Lexical Chain, TF-IDF, TextRank on both the benchmark datasets of BBC News Dataset and CNN News Dataset.

On the basis of our observation and analysis, we proposed the following extension which can be incorporated to further improve the efficiency and accuracy of automatic summary of news articles: One can combine several sentence scoring algorithms into one single system which will incorporate the diverse features of these algorithms to produce a better summary for news articles.

References

[1] Kumar, Akshi, Aditi Sharma, and Anand Nayyar. "Fuzzy Logic based Hybrid Model for Automatic Extractive Text Summarization." In *Proceedings of the 2020 5th International Conference on Intelligent Information Technology*, pp. 7-15. 2020.

[2] Gambhir, Mahak, and Vishal Gupta. "Recent automatic text summarization techniques: a survey." *Artificial Intelligence Review* 47, no. 1 (2017): 1-66.

[3] VS, Raj Kumar, and D. Chandrakala. "A survey on text summarization using optimization algorithm." *ELK Asia Pacific Journal of Computer Science and Information Syatems* 2, no. 1 (2016): 31-40.

[4] Sharma, Richa, and Prachi Sharma. "A survey on extractive text summarization." *International Journal* 6, no. 4 (2016).

[5] Mihalcea, Rada, and Paul Tarau. "Textrank: Bringing order into text." In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404-411. 2004.

[6] Sethi, Prakhar, Sameer Sonawane, Saumitra Khanwalker, and R. B. Keskar. "Automatic text summarization of news articles." In *2017 International Conference on Big Data, IoT and Data Science (BID)*, pp. 23-29. IEEE, 2017

[7] Kosmajac, Dijana, and Vlado Kešelj. "Automatic Text Summarization of News Articles in Serbian Language." In *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pp. 1-6. IEEE, 2019.

[8] Allahyari, Mehdi, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. "Text summarization techniques: a brief survey." *arXiv preprint arXiv:1707.02268* (2017).

[9] Gaikwad, Deepali K., and C. Namrata Mahender. "A review paper on text summarization." *International Journal of Advanced Research in Computer and Communication Engineering* 5, no. 3 (2016): 154-160.

[10] Prathima, M. R., and H. R. Divakar. "Automatic Extractive Text Summarization Using K-Means Clustering." *International Journal of Computer Sciences and Engineering* (2018).

[11] Malhotra, Shilpi, and Ashutosh Dixit. "An effective approach for news article summarization." *International Journal of Computer Applications* 76, no. 16 (2013).

[12] Moratanch, N., and S. Chitrakala. "A survey on extractive text summarization." In *2017 international conference on computer, communication and signal processing (ICCCSP)*, pp. 1-6. IEEE, 2017.

[13] Kallimani, Jagadish S., K. G. Srinivasa, and B. Eswara Reddy. "Summarizing news paper articles: experiments with ontology-based, customized, extractive text summary and word scoring." *Cybernetics and Information Technologies* 12, no. 2 (2012): 34-50.

[14] T. Ahmad and N. Ahmad, "A Simple Guide to Implement Data Retrieval through Natural Language Database Query Interface (NLDQ)," 2019 8th International Conference System

ISSN: 2005-4262IJGDC

Modeling and Advancement in Research Trends (SMART), Moradabad, India, 2019, pp. 37-41, doi: 10.1109/SMART46866.2019.9117501.

[15] Akhtar, Nadeem, Hira Javed, and Tameem Ahmad. "Hierarchical summarization of text documents using topic modeling and formal concept analysis." In Data Management, Analytics and Innovation, pp. 21-33. Springer, Singapore, 2019.

[16] Kauser, Sadia, Ayesha Rahman, Asad Mohammed Khan, and Tameem Ahmad. "Attribute-based access control in web applications." In Applications of Artificial Intelligence Techniques in Engineering, pp. 385-393. Springer, Singapore, 2019.

[17] P. Bansal, Somya, N. Kamaal, S. Govil, and T. Ahmad, "Extractive review summarization framework for extracted features," Int. J. Innov. Technol. Explor. Eng., 2019.

[18] Fang, Wei, et al. "A method of automatic text summarisation based on long short-term memory." International Journal of Computational Science and Engineering 22.1 (2020): 39-49.

[19] Orăsan, Constantin. "Automatic summarisation: 25 years On." Natural Language Engineering 25.6 (2019): 735-751.

[20] Mohamed, Muhidin, and Mourad Oussalah. "SRL-ESA-TextSum: A text summarization approach based on semantic role labeling and explicit semantic analysis." Information Processing & Management 56.4 (2019): 1356-1372.

[21] Mohd, Mudasir, Rafiya Jan, and Muzaffar Shah. "Text document summarization using word embedding." Expert Systems with Applications 143 (2020): 112958.

[22] Ferreira, Rafael, et al. "Assessing sentence scoring techniques for extractive text summarization." Expert systems with applications 40.14 (2013): 5755-5764.

[23] Bidoki, Mohammad, Mohammad R. Moosavi, and Mostafa Fakhrahmad. "A semantic approach to extractive multi-document summarization: Applying sentence expansion for tuning of conceptual densities." Information Processing & Management 57.6 (2020): 102341.

[24] Mackey, Andrew, and Israel Cuevas. "Automatic text summarization within big data frameworks." Journal of Computing Sciences in Colleges 33.5 (2018): 26-32.

[25] Liu, Yike, et al. "Graph summarization methods and applications: A survey." ACM Computing Surveys (CSUR) 51.3 (2018): 1-34.

[26] Zhou, Xinyi, and Reza Zafarani. "A survey of fake news: Fundamental theories, detection methods, and opportunities." ACM Computing Surveys (CSUR) 53.5 (2020): 1-40.

[27] Ahmad, Tameem, et al. "Beginning with exploring the way for rumor free social networks." Journal of Statistics and Management Systems 23.2 (2020): 231-238.

[28] T. Ali, T. Ahmad and M. Imran, "UOCR: A ligature based approach for an Urdu OCR system," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2016, pp. 388-394.

ISSN: 2005-4262IJGDC

ISSN: 2005-4262IJGDC