STM JOURNALS

Journal of
Artificial Intelligence
Research & Advances

ISSN: 2395-6720
Volume 11, Issue 2, 2024
May–August
DOI Journal: 10.37591/JoAIRA

https://journals.stmjournals.com/joaira

Review JoAIRA

Extractive Text Summarization: An Application Based Study

Deepanshu Anand[1,*], Yugansh Gupta[1], Arnav Sabharwal[1],
Vinay Kumar Saini[2], Anshu Khurana[3]

Abstract

Text summarization is an essential tool for extracting important information from lengthy texts or documents. Text Summarization has two main methodologies namely: Extractive Summarization and Abstractive Summarization. This study concentrates on extractive summarising, which selects significant sentences straight from the source material to create a summary. It is a popular option for many practical applications since it frequently produces summaries that are more accurate in terms of substance. In abstractive summarization, the summaries are generated by using the words that are not in the original text. However, the disadvantage of the above technique lies in the areas, where we want to retain the original text from the source. Hence the need of extractive summarization arises. Although, there are few drawbacks associated to extractive summarization which include the possibility of repetition and the dependence on pre-existing content. This study investigates how extractive summarization could be used to create end-to-end applications that are broadly applicable across various applications like legal document analysis. Through the resolution of these constraints and the utilisation of advances in natural language processing, extractive summarization could provide beneficial outcomes for a range of applications.

Keywords: Text summarization, supervised learning, unsupervised learning, semantics, extractive summary

INTRODUCTION

In recent times, the need for text summarization arises from the ever-increasing amount of information available in various forms such as news articles, research papers, social media posts, and more [1]. With the availability of such a vast amount of information, it is becoming increasingly difficult for people to keep up with the information overload. Through the provision of a succinct and easily readable summary of the original content, text summarization aids in resolving this issue. It saves readers, who might not have the time or desire to read the entire text, time and effort [1]. A variety of applications, including news summarization [2], document summarization [1], legal document summarization [3], and others, use text summarization. Text summarization is helpful for individuals, but it also has uses in fields like finance, law, and healthcare where there is a need to process and analyse massive amounts of text. In general, text summarization has developed into a crucial tool for organizing and making sense of the enormous volumes of information we come across every day.

***Author for Correspondence**
Deepanshu Anand
E-mail: deepanshu0810@gmail.com

[1]Student, Department of Information Technology, Maharaja Agrasen Institute of Technology, Delhi, India
[2]Associate Professor, Department of Information Technology, Maharaja Agrasen Institute of Technology, Delhi, India
[3]Assistant Professor, Department of Information Technology, Maharaja Agrasen Institute of Technology, Delhi, India

Received Date: May 24, 2024
Accepted Date: June 06, 2024
Published Date: July 10, 2024

Citation: Deepanshu Anand, Yugansh Gupta, Arnav Sabharwal, Vinay Kumar Saini, Anshu Khurana. Extractive Text Summarization: An Application Based Study. Journal of Artificial Intelligence Research & Advances. 2024; 11(2): 41–48p.

© STM Journals 2024. All Rights Reserved

In extractive summarization, the most significant sentences or phrases from the original text are chosen, extracted, and then presented as a summary [4]. In order to choose the most pertinent content, this type of summarizing often entails identifying crucial words, phrases, or sentences. Compared to abstract summarization, this method is simpler and easier to apply [5]. Additionally, it frequently produces summaries that are more accurate in terms of information but may not have the same consistency and originality as summaries produced by humans.

Abstractive summarizing is creating a summary that incorporates new words and phrases while retaining the main ideas and information of the original text. It is complex and requires more sophisticated technology, but it has the potential to produce summaries that are more legible, cohesive, and resemble human-generated summaries in style.

The choice of technique depends on the particular demands and requirements of the task or application. Both extractive and abstractive summarization offer benefits and drawbacks. To provide summaries that are more useful, certain summarizing systems may mix the two methods.

Some of the key areas where we could apply text summarization are news [2] and media: automatic summarization of new stories could be used to speed up information dissemination and improve user experience. Another important area is legal document analysis [3]: summarization could help in managing massive case files and could be used to identify pertinent precedents, essential arguments, or crucial evidence. Search engines frequently show excerpts or brief summaries next to search results. By letting consumers select which search results to click on depending on the summary offered; summarization helps to increase the relevance of search results. Using text summarization in business, decision makers may quickly comprehend market trends, competitor activity, customer opinion, and other pertinent information thanks to summaries, which facilitate strategic planning and well-informed decision making.

In this study, we will use extractive text summarizing methods to develop a whole application. In this application, the most significant sentences or phrases from the original text are picked out and presented in a summary. In comparison to abstractive summarization, this method is simpler and easier to put into practice. It is a popular option for many practical applications since it frequently produces summaries that are more accurate in terms of substance. By concentrating on extractive summarization, we hope to give a thorough analysis of the available methods and pinpoint potential areas for efficiency and effectiveness growth. This methodology will help understand extractive text summarization and its applications aspect.

LITERATURE SURVEY

In the area of natural language processing known as inferred text summarization, the most significant sentences or phrases are taken out of the data to provide a summary. Since there has been a lot of recent study on this subject, this literature review will concentrate on the key findings from 2000 to 2020. The most significant sentence is chosen using the Centroid approach based on the similarity score between each sentence and the entire manuscript. However, it was discovered that the centroid method was inefficient and unable to generate precise places.

In 2002, Luhn proposed a method called "AutoSummENG" that uses statistical data and annotations to identify the most important sentences in a document [6]. The AutoSummENG method achieves better results than the centre of gravity method, but its ability to generate both simple and informative points is still limited. In 2004, the study by Erkan and Radev published a method called "LexRank", which uses a technique similar to PageRank to identify the most important articles in a document [7]. LexRank surpasses the centre of gravity and AutoSummENG methods and becomes a model for future research in this area.

© STM Journals 2024. All Rights Reserved

In 2004, the study by Mihalcea and Tarau proposed a method called "TextRank", which is similar to LexRank but uses a different scheme for sentences in the document [8]. TextRank achieved similar results to LexRank and became a popular method for text analysis.

The DUC (Document Understanding Conference) abstraction study was launched in 2011, which provides a standardized benchmark for abstraction systems. Various systems have been developed for the DUC task, including MEAD, a multi-data aggregator developed by Radev and colleagues that combines TextRank with the centroid method to create quality content [7].

In 2015, Nallapati *et al*. proposed a method called "SummaRuNNer" that uses neural networks to learn the significance of sentences in the data [9]. SummaRuNNer summarizes the most recent results of DUC-2004 and DUC-2007 data and paves the way for future research using neural networks for feature extraction.

In 2017, Cheng and Lapata proposed a method called "neural sentence sequencing", which uses a neural network to learn the suggestion of selected sentences in context [10]. The methodology complements the case results of the DUC-2004 and DUC-2007 data and demonstrates the importance of sentence ordering in creating good content.

In 2019, Liu proposed a method called "Fine-tuned BERT-based Summarization" which fine-tunes a pre-trained BERT model for the summarization task [11]. The method achieves state-of-the-art results across multiple datasets, demonstrating the importance of fine-tuning before language training for feature extraction.

In conclusion, extractive text summarization has advanced significantly in recent years, thanks in large part to the research network. The industry has witnessed the development of various benchmark datasets, the development of neural community-based and graph-based approaches, and the proof of the efficacy of trained language models.

LEXRANK AND TEXTRANK

Two well-liked algorithms for extractive text summarization are LexRank and TextRank, as shown in Figures 1 and 2. Although there are numerous similarities between the two algorithms, there are a few significant distinctions as well:

Both LexRank and TextRank are graph-based algorithms, however their methods for building the underlying graph are different. Typically employing TF-IDF or other sentence embeddings, LexRank creates a similarity graph of sentences based on the cosine similarity of their vector representations. Contrarily, TextRank builds a graph by treating phrases or words as nodes and connecting them depending on how frequently they occur together within a text window.

LexRank and TextRank rank the significance of sentences in a document using various scoring systems. By modifying Google's PageRank algorithm, LexRank rates each sentence according to how central it is to the graph. Higher marks are awarded to sentences that are similar to other significant sentences. The significance of a sentence is evaluated by the total of the scores of its nearby sentences in the graph in TextRank, which uses a straightforward iterative process.

Sentence length matters because LexRank tends to be more capable of coping with fluctuations in sentence length. It accounts for sentence length while calculating similarity scores, which reduces its inclination to favour shorter or longer sentences. Sentence length is not explicitly taken into account in TextRank, therefore longer sentences may do worse in the ranking process.

Evaluation and performance: Based on the dataset and evaluation metrics utilised, the performance of LexRank and TextRank has been compared and evaluated in a number of researches. Although both algorithms have demonstrated their efficacy in extractive summarization tasks, the specific performance may differ depending on the dataset's features and the implementation choices made.

© STM Journals 2024. All Rights Reserved

The graph-based text summarization algorithms LexRank and TextRank are just two examples; various modifications and expansions of these algorithms have been presented in the literature.

METHODOLOGY

This section outlines the methodology employed to develop a Text Summarization web app. The purpose of the study is to determine the effectiveness of three Text Summarization algorithms: a custom LexRank algorithm, built-in LexRank Algorithm and Sumy Library TextSummarizer (TextRank). The Web Application allows the user to upload a .txt file which is then processed by and three summaries are given to the user. Python Flask Framework is used to develop the web application.

Data Processing

Since we have attempted to develop a web application, our model is solely dependent on the data provided by the user. The file uploaded by the user is stored on server-side and then used for preprocessing. The preprocessing involves steps to remove any unwanted sequence of characters like blank spaces and reference numbers from the text. This could be easily done by regex (regular expression). After this, the formatted text is then sent to the model.

Working of Models

The formatted data is then passed through the tokenizer in which the sentences from the text are converted into vectors and then a similarity matrix is created. The process of creating a similarity matrix depends upon the type of model we are using; in our case we are using three models, so there are three processes of creating a similarity matrix as shown in Figures 3 and 4.

In the first model i.e., TextRank model, the similarity score is calculated using Bag-Of-Words and in the next model i.e., LexRank algorithm, it uses TF-IDF scores and cosine similarity to determine the similarity between sentences.

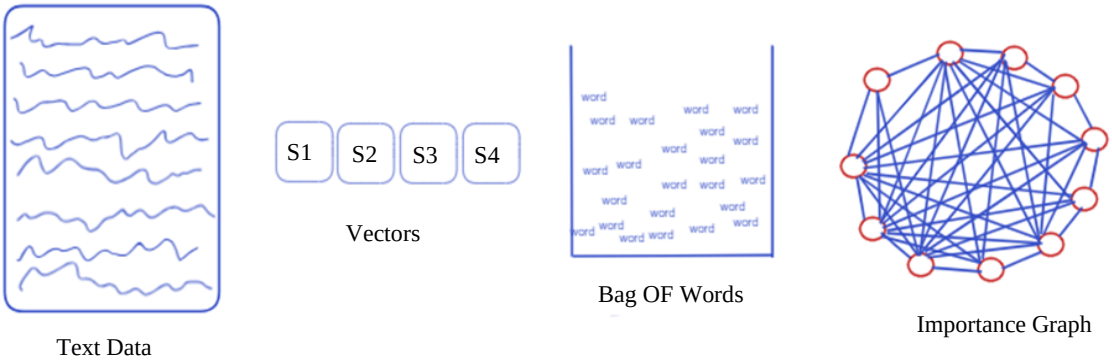

Figure 1. Sequence of actions in TextRank.

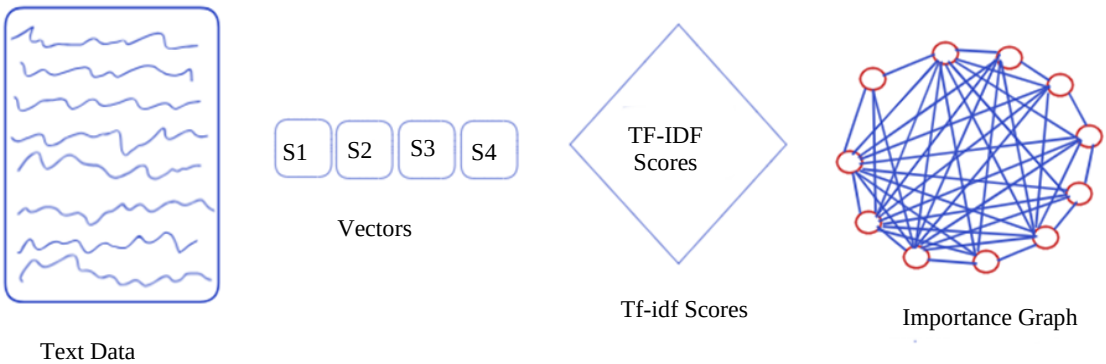

Figure 2. Sequence of actions in LexRank.

© STM Journals 2024. All Rights Reserved

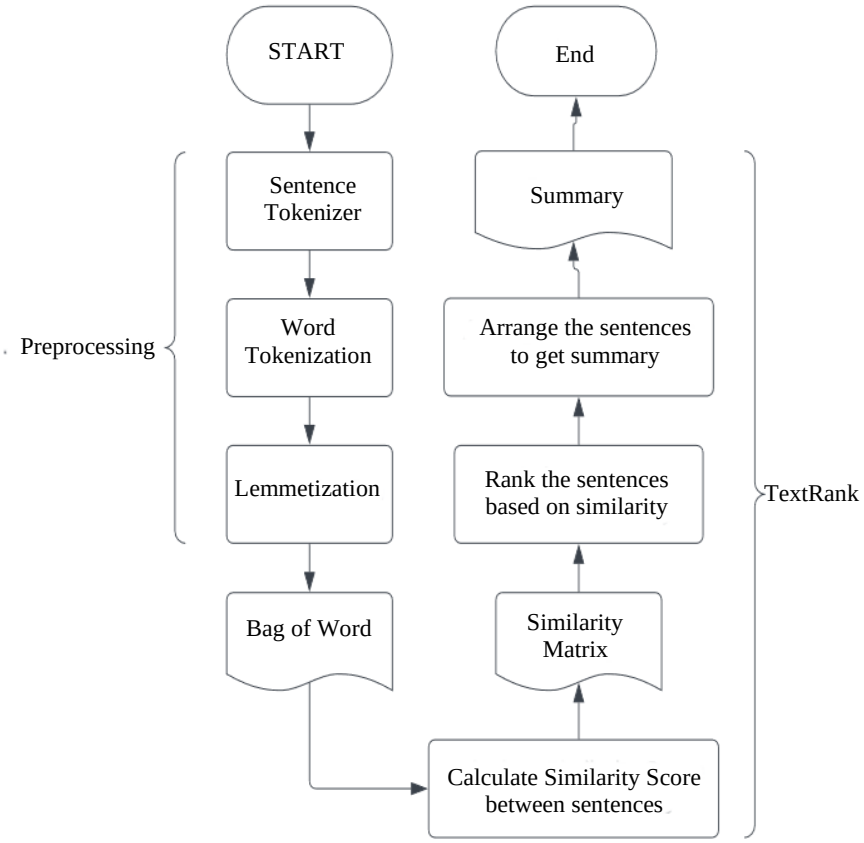

Figure 3. TextRank Flowchart.

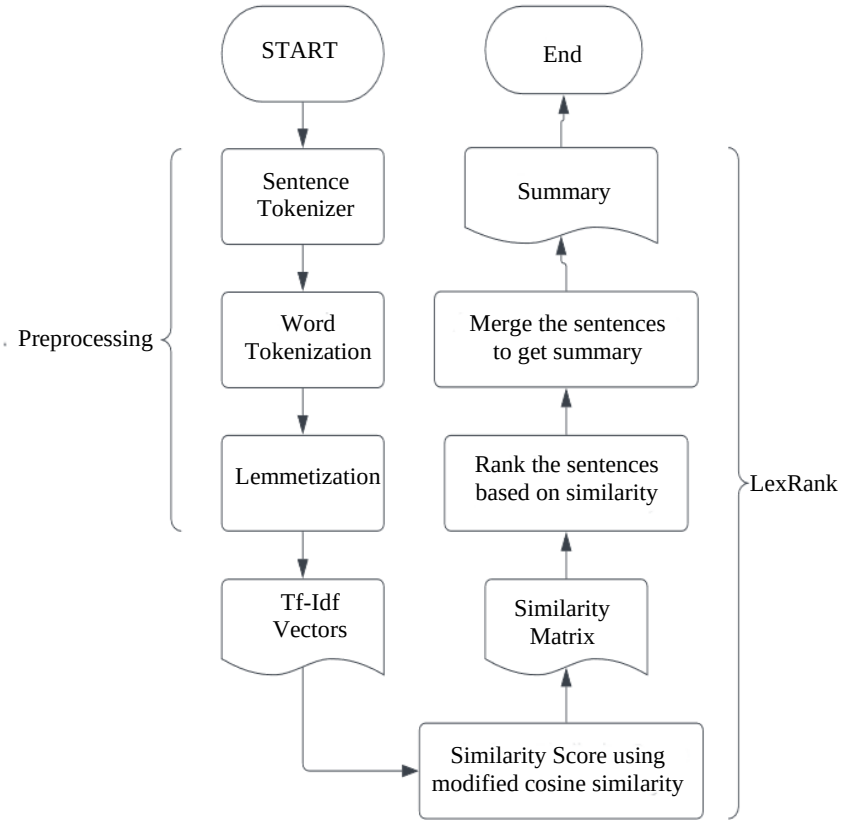

Figure 4. LexRank Flowchart.

© STM Journals 2024. All Rights Reserved

$$TextRank\ similarity\ Score = \frac{no\ of\ words}{\log|s1|+\log\ |s2|} \quad (1)$$

$$\cos(x,y) = \frac{\sum_{w\in x,y} tf_{w,x} tf_{w,y} (idf_w)^2}{\sqrt{\sum_{x_i\in x}\left(tf_{x_i x} idf_{xi}\right)^2}\sqrt{\sum_{y_i\in y}\left(tf_{y_i y} idf_{yi}\right)^2}} \quad (2)$$

Evaluation

The evaluation metrics used in this application is Rouge Score (ROUGE is short for Recall-Oriented Understudy for Gisting Evaluation). It is a tool to automatically assess the quality of the summaries generated by computer by contrasting it to reference summaries [12]. This metrics calculates the number of overlapping units i.e. word pairs, n-grams or sequence of words between the generated summary and the reference summary.

There are majorly three types of ROUGE scores:

- ROUGE-N: measures n-gram overlap between two texts.
- ROUGE-L: measures the recall between longest matching sequence.
- ROUGE-S: measures the recall of word-pairs having a definite gap between their occurrences.

In our analysis of different models, we are using the following variants of the ROUGE score:

- ROUGE-1: precision, recall and f1-score is calculated by using the number of unigrams in generated text that appear in the reference text.
- ROUGE-2: precision, recall and f1-score is calculated by using the number of di-grams in generated text that also appear in reference text.
- ROUGE-L: metrics calculated using the longest common subsequence between the two texts.

With respect to the user application, the evaluation is challenging. Since it is a user application and the data that user uploads are not necessarily a standard data, here occurs the biggest challenge in validating the summaries. To validate a summary or to calculate accuracy of the summary we need a validated summary to compare the results, but in our case, the data is new to the model and it does not have any validation set. Also, the idea of calculating the correlation of the generated summaries with the text document to evaluate them is also not useful in this case because it will always return a correlation of 1 because the summaries have the exact sentences from the original text.

In text summarization a validation set is created by expert annotation and crowd sourcing. Since the data we are working on is completely new and random as per the model. A proposed solution for this problem could be considering a widely accepted pre-trained model to generate summaries. The summaries generated by that model could be considered as the validation summary and then we could evaluate summaries.

RESULTS

We used PubMed dataset to evaluate the models used in the user application. We report Rouge-1/2 and Rouge-L scores as shown in Table 1.

The result for PubMed dataset is low because the generated summaries are evaluated against paraphrased summaries, thus, the number of overlapping words will be comparatively less because the generated summaries are extractive in nature, i.e. generated by using same sequence of words that exist in the original text.

Table 1. Report for Rouge-1/2 and Rouge-L scores.

Model	Rouge-1	Rouge-2	Rouge-L
TextRank	38.66	15.87	34.53
LexRank	39.19	13.89	34.59

© STM Journals 2024. All Rights Reserved

Table 2. Report for Rouge-L scores.

Model	Rouge-L
TextRank	41.87
LexRank	42.18

We use another dataset FacetSum to evaluate the models and we report Rouge-L scores (Table 2).

The results for this dataset are better than the previous because the FacetSum dataset contains extractive summaries to evaluate unsupervised models.

CHALLENGES IN THIS APPROACH

In machine learning, no model is perfect, every model has its limitation; similarly, Extractive Text Summarization also has some limitations. The most common and biggest limitation of Extractive Text Summarization is that it brings semantic ambiguity. In the summaries generated by Extractive Text Summarizer, there are exact same sentences from the original text. The most important sentences are considered from different parts of the original text and combined together and it is not necessary that the sentences after combining follow the semantics and the language rules and might not make complete sense sometimes. This problem is solved by Abstractive Text Summarization; in this approach the summaries are generated in a completely paraphrased form thus generating semantically correct summaries.

CONCLUSION

The area of extractive text summarization has been examined in this research work, with an emphasis on two well-known methods: Lexrank and Textrank. We talked about their underlying algorithms, which use graph-based methods to recognize crucial phrases in a text document and produce succinct summaries. Numerous applications, including news and media, information retrieval, document management, social media analysis, legal research, business intelligence, healthcare, education, and more, have shown that these strategies are beneficial. Additionally, we demonstrated a complete application built using Flask and Python that enables users to upload text files and produce summaries using the implemented Lexrank and Textrank algorithms. The web application gives users quick and precise summaries of extensive documents, serving as a realistic example of how these techniques might be incorporated into real-world systems.

We also talked about the difficulties with extractive text summarization throughout the study. Maintaining summary coherence, handling terminology relevant to a certain subject, navigating various text formats, dealing with information redundancy, and adjusting to changing linguistic patterns are some of these difficulties. Future studies in this area should focus on overcoming these difficulties by utilizing cutting-edge natural language processing methods, adding domain-specific information, and investigating cutting-edge strategies to raise the caliber and relevancy of extraction summaries.

Future Scope

The current research has shed light on various aspects of extractive text summarization and its applications. However, there are still several avenues for future exploration and improvement. The following areas offer promising directions for future research:

Investigate and develop more sophisticated algorithms and techniques for extractive text summarization that go beyond traditional approaches like LexRank and TextRank.

Explore deep learning models, neural networks, and other advanced machine learning techniques to enhance the quality and accuracy of the generated summaries.

Incorporate domain-specific knowledge and ontologies to improve the relevance and domain-specific understanding of the summaries. Explore hybrid approaches that combine extractive and abstractive

© STM Journals 2024. All Rights Reserved

summarization techniques to generate more informative and coherent summaries. Extend the current research to address the challenges of multilingual and cross-lingual text summarization. Investigate methods to effectively summarize documents in languages other than English, considering language-specific characteristics and nuances.

The most important area which is left behind in this study is the evaluation of newly introduced data in the model, further exploration and research is required to find out a way to give user a metric along with the summaries so that he/she could choose the best summary.

REFERENCES

1. El-Kassas WS, Salama CR, Rafea AA, Mohamed HK. Automatic text summarization: A comprehensive survey. Expert Syst Appl. 2021; 165: 113679.
2. Sethi P, Sonawane S, Khanwalker S, Keskar RB. Automatic text summarization of news articles. In 2017 IEEE International Conference on Big Data, IoT and Data Science (BID). 2017 Dec; 23–29.
3. Kanapala A, Pal S, Pamula R. Text summarization from legal documents: a survey. Artif Intell Rev. 2019; 51(1): 371–402.
4. Moratanch N, Chitrakala S. A survey on extractive text summarization. In 2017 IEEE international conference on computer, communication and signal processing (ICCCSP). 2017 Jan; 1–6.
5. Moratanch N, Chitrakala S. A survey on abstractive text summarization. In 2016 IEEE International Conference on Circuit, power and computing technologies (ICCPCT). 2016 Mar; 1–7.
6. Luhn HP. The Automatic Creation of Literature Abstracts. *IBM J Res Dev*. 1958 Apr; 2(2): 159–165. doi: 10.1147/rd.22.0159.
7. Erkan G, Radev DR. Lexrank: Graph-based lexical centrality as salience in text summarization. J Artif Intell Res. 2004; 22(1): 457–479.
8. Mihalcea R, Tarau P. Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing. 2004 Jul; 404–411.
9. Nallapati R, Zhai F, Zhou B. Summarunner: A recurrent neural network-based sequence model for extractive summarization of documents. In Proceedings of the AAAI conference on artificial intelligence. 2017 Feb; 31(1).
10. Cheng J, Lapata M. Neural summarization by extracting sentences and words. arXiv preprint arXiv:1603.07252. 2016.
11. Liu Y. Fine-tune BERT for extractive summarization. arXiv preprint arXiv:1903.10318. 2019.
12. Lin CY. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out. Barcelona, Spain: Association for Computational Linguistics; 2004 Jul; 74–81.

© STM Journals 2024. All Rights Reserved