UDC 004.912 **doi: 10.32620/reks.2023.2.01**

Md. Ahsan HABIB[1], Romana Rahman EMA[1,*], Tajul ISLAM[2],
Md. Yasir ARAFAT[1], Mahedi HASAN[1]

[1] ***Department of Computer Science and Engineering, Jashore University of Science and Technology, Jashore-7408, Bangladesh***
[2] ***Department of Computer Science and Engineering, North Western University, Bangladesh***

AUTOMATIC TEXT SUMMARIZATION BASED ON EXTRACTIVE-ABSTRACTIVE METHOD

*The choice of this **study** has a significant impact on daily life. In various fields such as journalism, academia, business, and more, large amounts of text need to be processed quickly and efficiently. Text summarization is a technique used to generate a precise and shortened summary of spacious texts. The generated summary sustains overall meaning without losing any information and focuses on those parts that contain useful information. The **goal** is to develop a model that converts lengthy articles into concise versions. The **task** to be solved is to select an effective procedure to develop the model. Although the present text summarization models give us good results in many recognized datasets such as cnn/daily- mail, newsroom, etc. All the problems can not be resolved by these models. In this paper, a new text summarization **method** has been proposed: combining the Extractive and Abstractive Text Summarization technique. In the extractive-based method, the model generates a summary using Sentence Ranking Algorithm and passes this generated summary through an abstractive method. When using the sentence ranking algorithm, after rearranging the sentences, the relationship between one sentence and another sentence is destroyed. To overcome this situation, Pronoun to Noun conversion has been proposed with the new system. After generating the extractive summary, the generated summary is passed through the abstractive method. The proposed abstractive model consists of three pre-trained models: google/pegusus-xsum, facebook/bart-large-cnn model, and Yale-LILY/brio-cnndm-uncased, which generates a final summary depending on the maximum final score. The following **results** were obtained: experimental results on CNN/daily-mail dataset show that the proposed model obtained scores of ROUGE-1, ROUGE-2 and ROUGE-L are respectively 42.67 %, 19.35 %, and 39.57 %. Then, the result has been compared with three state-of-the-art methods: JEANS, DEATS and PGAN-ATSMT. The results outperform state-of-the-art models. Experimental results also show that the proposed model is qualitatively readable and can generate abstract summaries. **Conclusion**: In terms of ROUGE score, the model outperforms some art-of-the-state models for ROUGE-1 and ROUGE-L, but doesn't achieve good result in ROUGE-2.*

Keywords: *Text Summarization; Extractive Summarization; Abstractive Summarization; Sentence Ranking Algorithm; Text Generation; Noun Pronoun Conversion.*

Introduction

Text summarization is a process used to generate a precise and shortened summary of spacious texts. The generated summary sustains overall meaning without losing any information and focuses on those parts that contain useful information [1]. It is a splendid approach that is used to reduce an article to its main concepts [2]. The purpose of the text summarization is to convert lengthy articles into concise versions. It can be helpful when we are short on time or when we need to find specific information in a text. For example, intelligent analysis systems of medical data are used for decision support in disease diagnosis [3]. If the process is performed manually, it could be difficult and costly to undertake. Overcoming this task is a significant step in understanding the natural language.

At present, everyone wants to access enormous amounts of information quickly. Huge amounts of text data are accessible online, which presents not only an opportunity but also a challenge. As a result, data being more readily available leads to data overload problems [4]. Social media calls for experts to process this flow of data carefully and attentively to release all relevant information that can be a subject for strategic monitoring [5]. In the modern era, an important task is to find and select information from a research article [6]. Most of the visited information is insignificant and redundant, and it may not maintain the desired meaning. When everyone needs a piece of specific information from an online news article, they must search through its content and alleviate the redundant information from the article. This process is complex and should spend most of the time finding the necessary information. Thus, extracting useful information using an automatic text summarizer that eliminates redundant and meaningless information is becoming important. Implementing an automatic text summarization can reduce the time spent researching information and enhance the readability of an article. It helps to find the necessary information in a short time [7].

Based on previous studies, the text summarization process can be divided into two classes [8]. The Extractive Text Summarization process is the first category that uses conventional systems, and this system generates a summary by cropping significant segments of the source article and combining those segments to produce an understandable summary [9]. The other category is the Abstractive Text Summarization process. This process generates a precise and compact summary that holds the principal concepts of the main article. The summaries are generated by the abstract method and contain new sentences and phrases that may not appear in the original article [10].

Over the last decades, researchers have proposed many extractive and abstractive text summarization approaches using various techniques. Although the present text summarization models give us good results in many recognized datasets such as cnn/daily- mail, newsroom, etc. All the problems can not be resolved by these models. There are two main essential factors to evaluate the text summarization model, namely semantic and syntactic structure [11]. These two different varieties of models focus on only one factor.

The extractive text summarizer produces a summary of the sentence in accordance with the source article. The disadvantages of the existing model are that the generated summary may not be meaningful sentence by sentence with respect to the main article. On the other hand, the advantages of abstractive text summarizer models summarize with semantic items [12]. After training, it creates a sequence of keywords based on the arrangement between the words. The disadvantage of the abstractive model, the sequence of keywords for syntactic structure is difficult to meet the requirement.

Therefore, this study aims to build a model of the text summarization process to generate a summary of an article. The research subjects are to determine models and methods of the text summarization process based on extractive and abstractive. To obtain the objective of the study, the following tasks have been formulated:

1. Informative sentences should be extracted using sentence ranking.
2. Extracted sentences should be analyzed for sentence-to-sentence relationships.
3. To overcome the sentence-to-sentence relationship problem, the pronoun of the sentence should be changed with the nearest noun of the sentence.
4. To generate a more readable and abstractive summary, extracted sentences should be passed through an abstractive method.

In this paper, section 1, namely the related work, provides the state-of-the-art of text summarization process methods and models. Section 2, namely Materials and Methods of Research, provides the preliminaries of Extractive and Abstractive development models, also provides the idea of Sentence Ranking using Google page rank algorithm. Section 3, namely Result and Discussion, describes the performance of the proposed model. Conclusions provide the outcome and future work of the investigations.

1. Related Work

S. Song et al. [9] introduced an LSTM-CNN-based ATS framework, ATSDL (Abstract Text Summarization Deep Learning). It can generate new sentences by investigating more fine-grained fragments of semantic phrases. ATSDL consists of two major stages. In the first stage, it picks out phrases from the main sentences. In the second stage, it generates shorthand and concise text summaries. Experimental results of the proposed framework show that the ATSDL framework outperforms the syntactic and semantic structure and achieves better results than state-of-the-art models.

L. Liu et al. [10] presented an abstractive text summarization method using an adversarial process. In this method, they trained a generative model named G. The generative model works as a reinforcement learning agent. It takes the input of the original text and generates a short summary. They also trained a discriminator model named D and built it. It attempts to differentiate between the original summary and the generated summary. Experimental results show that this proposed model obtains better ROUGE scores than state-of-the-art models on cnn/daily-mail dataset.

A. Barrera et al. [13] introduced an abstractive text summarization framework. This framework model was based on the encoder-decoder model with a sequence-to-sequence oriented decorated with a deep recurrent generative decoder (DGRN). It generates abstractive summarization based on both discriminative deterministic and generative latent variables state. Experimental results on some datasets show that DGRN framework outperforms some benchmark methods.

K. Yao et al. [14] proposed an automatic text summarization technique based on an abstractive method. In this paper, they used a dual encoding technique. The dual encoder consists of primary and secondary encoders. The primary encoder regularly operates frieze encoding. On the other hand, a secondary encoder creates better fine encoding depending on the input original text. Finally, the two levels of encoding are merged and passed into the decoder, generating a more variant summary. The experimental results on some benchmark datasets (cnn/dailymail and DUC 2004) show that the proposed model performs better than the existing models.

T. Cai et al. [15] presented a text summarization model called the RC- Transformer (RCT). They added an extra RNN encoder to extend the transformer. The extended transformer captures the sequence-to-sequence context representations and generates a module to filter those contexts with local significance. The experimental results of the model show that it achieves better performance than some benchmark models.

In this case, to solve the existing problem and increase the system's accuracy, combining the extractive and abstractive text summarization models has been proposed. The main contribution of the investigation is the method that allowed overcoming sentence-to-sentence relationship problems. This provides:

– sentence extraction using a text ranking algorithm;

– overcoming sentence-to-sentence relationship problems in sentence extraction using a pronoun-to-noun conversion process.

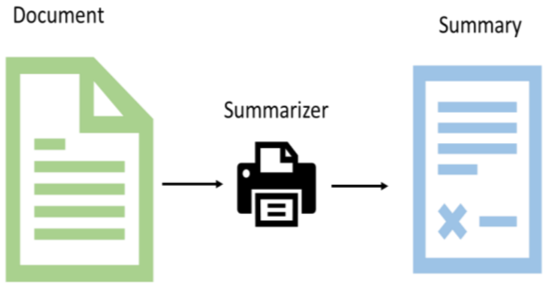


Fig. 1. Text Summarization Process

First, the system generates a partial summary of the source article based on sentence extraction and the text rank algorithm. Then, the generated partial summary passes through the abstractive based model. The abstractive model generates a meaningful final summary. In short, the overall system is shown in Fig. 1.

2. Materials and Methods of Research

2.1. Preliminaries

2.1.1. Extractive Text Summarization

Extractive-based text summarization is a technique that selects a few sentences from the original text to create a summary [16]. It can be very accurate as it simply identifies the most important sentences in a text. It can be helpful when we need to create an accurate summary and reliable. First, the intermediate representation was created using an extractive method. The major task of the representation is to collect the most significant information from the source text. Using sentence ranking, the extractive summarization is shown in Fig. 2.

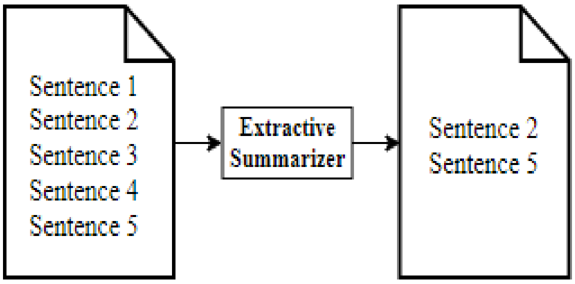


Fig. 2. Extractive Summarization

Text Rank Algorithm

In order to the google page ranking algorithm [17],

$$P(V_i) = (1 - d) + d * \\ * \sum_{j \in \int In\,(V_i)} \frac{1}{|Out(V_j)|} P(V_j), \qquad (1)$$

where P(Vi) represents the subject node score and P(Vj) represents all outgoing edges to node Vi.

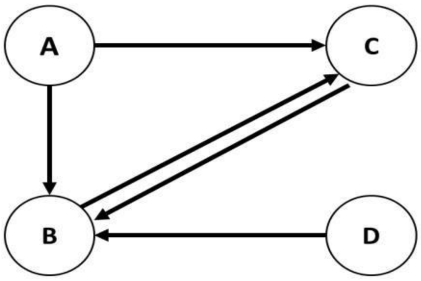


Fig. 3. Page Rank Graph [18]

The page rank graph is shown in Fig. 3. In this graph, a user starting at point A now goes to both C and B. So, the probability of going to B and C is ½. Then, starting at B, the user can go to only C. So, the probability of going from B to C is 1. In equation 1, d represents the damping factor. It also incorporates randomness in the page-ranking algorithm, and 1-d represents the user's move to another webpage. Generally, the damping factor is set to 0.85.

We have seen that the graph of the page rank algorithm is unweighted. For the text rank algorithm, it would not carry the full importance dividing with the out-degree. Thus, the graph and the equation are modified to a weighted graph. As a result, the equation becomes:

$$WP(V_i) = (1 - d) + d * \\ * \sum_{j \in \int In\,(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WP(V_j), \qquad (2)$$

where, WP(V_i) represents the weight of sentence, In (Vi) represents all ingoing edge from sentences(nodes) Vi, Out (Vj) represents all outgoing edge from sentences(nodes) Vj and the Wji or Wjk represents the weight factor of edge.

Sentence Extraction Task

Sentence extraction [17, 19] is a type of technique that is used for automatic summarization of a text. It identifies the most important sentences in a text using statistical heuristics. This approach is less expensive because it does not require any additional knowledge bases. To generate a graph for sentence ranking, the text rank algorithm creates a vertex for all sentences that will appear in the text. Then, the vertex is added to the generated graph. The co-occurrence system cannot be applied because of large sentences. So, we use a "similarity" between two sentences. To connect two sentences, the similarity relation is used [20]. To measure similarity, we use content overlap. The similarity of the two sentences is based on the number of tokens that are a common word and that word is present in the two sentences. The similarity between two sentences is given by:

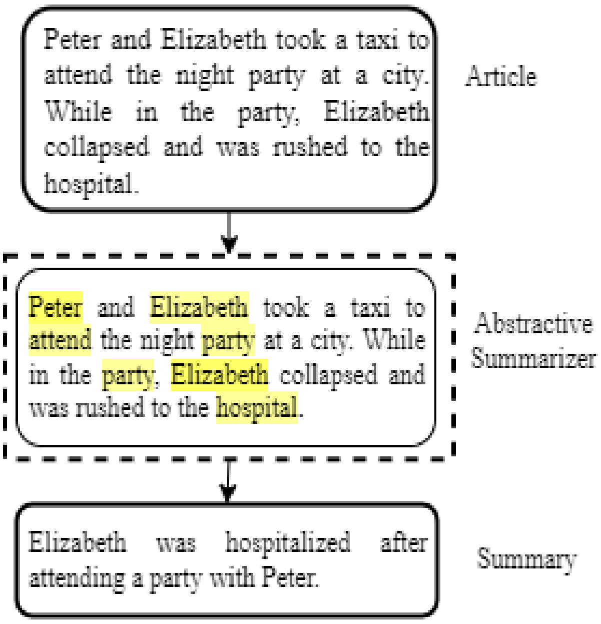
$$\text{Similarity}(P_i, P_j) = \frac{|\{W_k \mid W_k \in P_i \& W_k \in P_j\}|}{\log(|P_i|) + \log(|P_j|)}, \qquad (3)$$

where P_i and P_j represents two sentences and those sentences being represented by the N_i words set that find in the sentence:

$$P_i = W_1^i, W_2^i, \ldots\ldots\ldots W_N^i. \qquad (4)$$

2.1.2. Abstractive Text Summarization

Abstractive summarization is the process of creating a summary from the main ideas of a text, rather than copying the most important sentences from the text verbatim [21] It is an important field of Data Mining and Natural Language Processing [22, 23] Instead of the extractive text summarizer, they create a paraphrasing of the main content of a given text. For creating paraphrasing, they use a vocabulary that is distinct from the main document. It can be more accurate than extractive summarization because it can identify the most important information in a text even if it is not explicitly stated. This task is exceptionally comparative to summarize what we do as a people. We make a semantic statement for the article in our brains. At that point, we choose words from our common lexicon suitable within the semantics, to generate short and concise summary. The summary generation process using the abstractive summarizer is illustrated in Fig. 4.



Peter and Elizabeth took a taxi to attend the night party at a city. While in the party, Elizabeth collapsed and was rushed to the hospital.

Peter and Elizabeth took a taxi to attend the night party at a city. While in the party, Elizabeth collapsed and was rushed to the hospital.

Elizabeth was hospitalized after attending a party with Peter.

Fig. 4. Abstractive Summarization

Pre-trained Model

The **google/Pegasus-xsum** [24] model is a pre-trained model. It is trained with sampled gap sentences that are ratios on both Huge News and C4. It is also trained for 1.5M, which was trained with 500k sample essential sentences. The model indiscriminately samples a gap sentence ratio between 15 % and 45 %. In the pre-trained model, significant sentences are sampled. The sample uses 20 % uniform noise to importance scores, and to encode newline characters, the sentence-piece tokenizer is modified.

The **facebook/bart-large-cnn** [25] model is also pre-trained by corrupting text with an arbitrary noising function. The model learns to reconstruct the original text. It is especially efficient when fine-tuned for text generation. It also works perfectly for understanding assignments.

The **Yale-LILY/brio-cnndm-uncased** [26] is a pre-trained model that estimates the probability of system-generated summaries more accurately [27].

2.1.3. ROUGE Evaluation

To calculate the score and find the difference between the two articles, ROUGE evaluation [28] metrics are used. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation [29]. It is software packages and a set of metrics. It is used for evaluating automatic text or article summarization software in NLP. The ROUGE matrices find the difference between an automatically generated summary and a reference summary [30]. Some evaluation matrices are available:

1. ROUGE-N: Measures unigram, bigram, trigram and higher order n-gram overlap between the reference and system generated summary[31].

– **ROUGE-1**: ROUGE-1 indicates to the overlap of uni-gram (each word) between the reference summary and system generated summary;

– **ROUGE-2**: ROUGE-2 indicates to the overlap of bi-grams between the reference and system generated summaries.

2. ROUGE-L: ROUGE-L indicates Longest Commong Subsequence related statistics [32].

2.2. Dataset

The cnn/daily-mail dataset [33] has been used, which is an English dataset. The dataset contains more than 300k unique news articles as written by CNN and Daily Mail journalists. Both the extractive and abstractive text summarizations are supported by the current version of the dataset. This version was created for abstractive question answering and created for machine comprehension and reading. It is a non-anonymized text summarization dataset. It has three features. One is id, which contains the URL from which the story was retrieved from. The second feature name is article and the other feature represents highlights. The article feature represents the text of news articles, used as the document to be summarized. The highlight feature represents the joined text of highlights with and around each highlight, which is the target summary. The dataset has three splits: train, test, and validation. The number of instances in the train split is 287133, test split is 11490, and validation split is 13368.

2.3. Experimental Setup

In this experiment, the Python Programming Language of version 3.10 was used. To develop and write the script, we used VS Code editor. To experiment with the results, the Lenovo IdeaPad 320 was used, which has 8GB RAM, 240GB SSD, and Intel corei3 2.00GH CP. In the pre-trained model, to tokenize the input text, the return_tensors='pt' argument has been used to return PyTorch tensors instead of a list of Python integers. The max_length=512 argument is used to set the maximum length of the input tokens to 512, which indicates the maximum length that the model can handle. The 'truncation=True' argument has been used to tokenize input text that truncates the input text if it exceeds the maximum length. To generate a summary, a minimum length of 80 and a maximum length of 120 arguments are used to set the minimum and maximum length of tokens of the generated summary. To decode the generated summary, the skip_special_tokens=True argument has been set up to remove any special tokens such as [CLS] (represents the first token in the input sequence), [SEP] (mark the end of a sentence), and [PAD] (the input sequence to have a fixed length) from the decoded summary.

2.4. Methodology

The key component of the proposed automatic text summarization model is the extraction of some important sentences from the source article, which are then used as input to generate the final summary. Therefore, the proposed model employs a representation framework that generates a summary of an article. First, the sentence extraction has been described from the original article, which is called the extractive-based method, and then the details are presented of the abstractive-based method.

2.4.1. Extractive Based Model

The extractive based approach comprises extraction of the most significant phrases and sentences from the main article. Then, it merges the top significant sentences to generate an extractive summary. Thus, in this task, each sentence and word of the generated summary actually involves the source article.

The flowchart of the extractive-based model is shown in Fig. 5 and the working procedure of the model has been explained using pseudocode 1. The steps for generating an extractive summary are given below:

– first step, concatenating all source article text;

– second, splitting the concatenated text into each individual sentence;

– third, for each and every sentence, finding out vector representation;

– next, calculating the similarity score between sentence vectors and then storing the score in a matrix table. The procedure of calculation of similarity score between two sentences and storing that score into matrix table have been given in pseudocode 2;

– then, converting the similarity matrix into a graph where each sentence represents a node and the similarity score represents an edge;

– next, ranking all sentences where the highest score sentence placed in top and lowest score sentence placed in bottom;

– finally, a certain number of top-ranking sentences will generate a final summary.

Sample Text: During the 2016 presidential election, Trump was the respective nominee of the Republicans. He is a businessman and television personality who served as the 45th President of the United States from 2017 to 2021. He won the election in a stunning upset, defeating Clinton in the electoral college despite losing the popular vote.

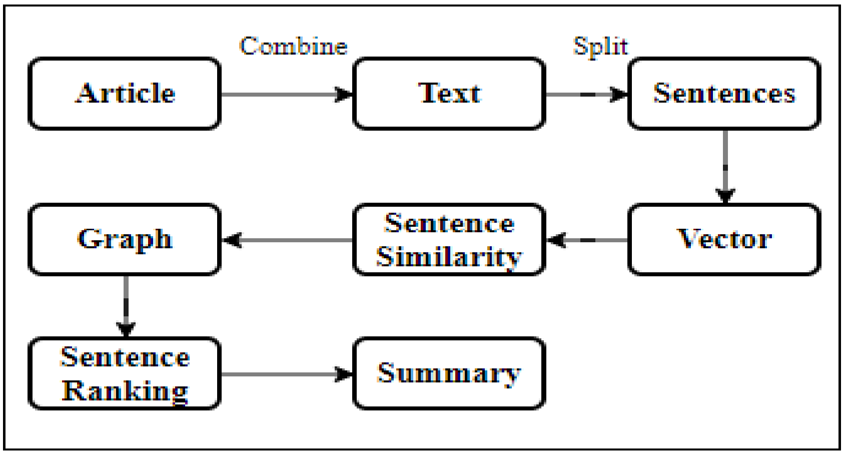


Fig. 5. Extractive Based Method [34]

Obama is also a well-known figure in American politics. He is a senior lecturer who focuses on issues such as healthcare reform, climate change, and foreign policy. He is better than him.

Pseudocode 1: Extraction Based Model

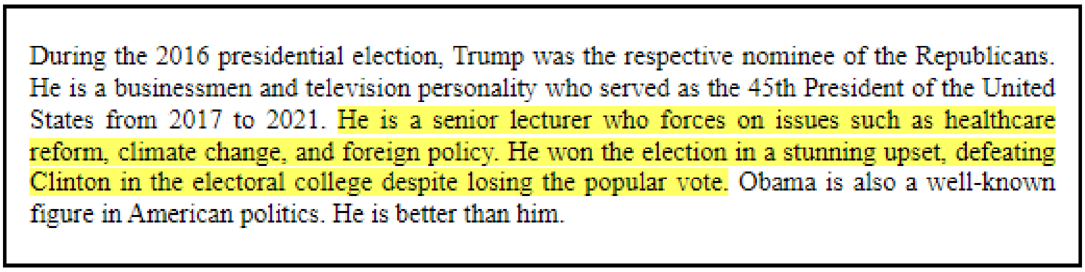
```
PROCEDURE generateExtractiveSummary(article, top n )
BEGIN PROCEDURE
    summarize_text← intilizize empty text
    sentences ← read_article
    sentence_similarity_matrix ← build_similarity_matrix(sentences)
    sentence_similarity_graph ← rank sentence in sentence_similarity_matrix
    score ← calculate the score in sentence_similarity_graph using equation no 2
    ranked_sentence← sort the rank and place top ranking sentences
    FOR i to top_n
        Add ranked_sentence[i] to summarize_text
    ENDFOR
    RETURN summarize_text
END PROCEDURE
```

Pseudocode 2: Similarity Matrix

```
PROCEDURE buildSimilarityMatrix(sentences)
BEGIN PROCEDURE
    n ← len(sentences)
    similarity_matrix← create n×n matrix with value 0
    For i=0 to n
        For j=0 to n
            similarity_matrix[i][j] = find sentence similarity of sentences[i] and sentences[j] using equatuion no 3
        ENDFOR
    ENDFOR
    RETURN similarity_matrix
END PROCEDURE
```

Sorting the sentences in the above sample text based on the highest score using the Text Rank algorithm (Sentence Ranking) leads to some situations which has been seen in Fig. 6.

A pronoun of a subject, in particular, represents the subject of the previous sentence. After sorting the sentence according to highest score, the subject-to-subject relationship are broken. To overcome this problem, the Pronoun to Noun conversion process has been added in Extraction Based Model which is illustrated in Fig. 7.



During the 2016 presidential election, Trump was the respective nominee of the Republicans. He is a businessmen and television personality who served as the 45th President of the United States from 2017 to 2021. He is a senior lecturer who forces on issues such as healthcare reform, climate change, and foreign policy. He won the election in a stunning upset, defeating Clinton in the electoral college despite losing the popular vote. Obama is also a well-known figure in American politics. He is better than him.

Fig. 6. After Applying Extractive Based Method

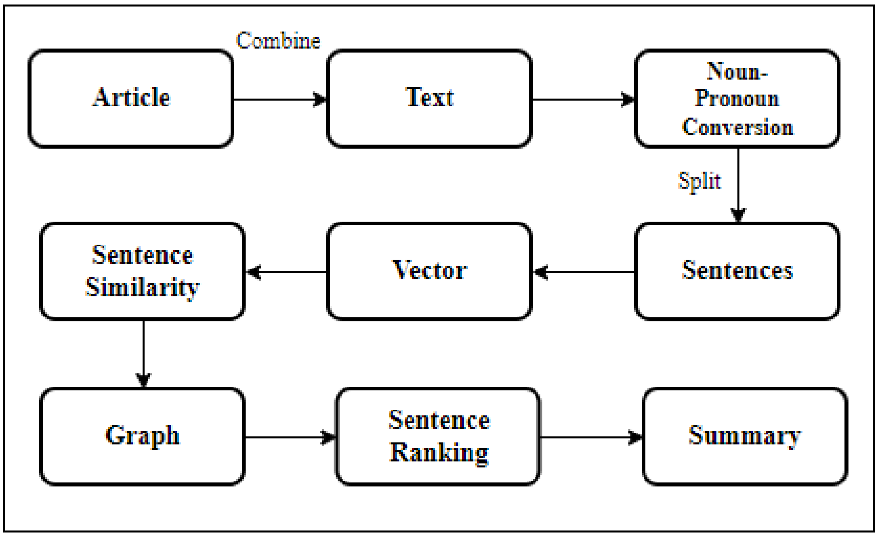


Fig. 7. Modified Extractive Based Method

The Pronoun to Noun conversion flowchart has been given in Fig. 8 and the working procedure of the flowchart has been stated using pseudocode 3.

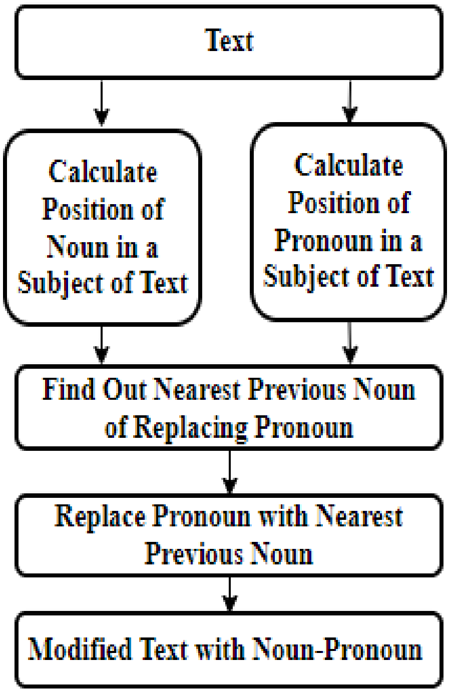

Text
Calculate Position of Noun in a Subject of Text
Calculate Position of Pronoun in a Subject of Text
Find Out Nearest Previous Noun of Replacing Pronoun
Replace Pronoun with Nearest Previous Noun
Modified Text with Noun-Pronoun

Fig. 8. Pronoun to Noun Conversion Process

The conversion step of Pronoun to Noun are given below in step by step:

– firstly, calculate the Noun position in a subject of a sentence of article;

– secondly, calculate the Pronoun position in a subject of a sentence of article;

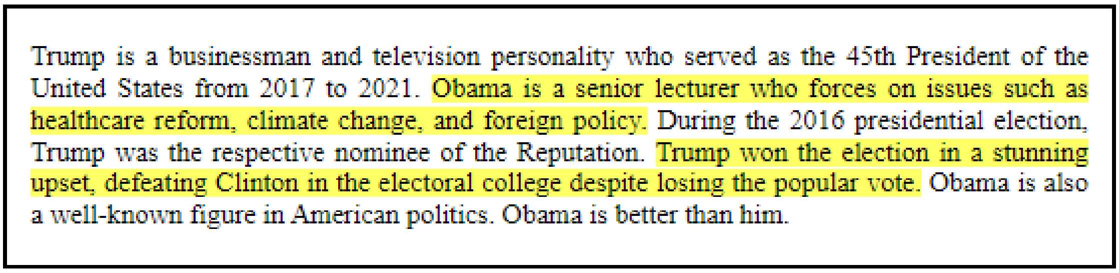
– then, find out the previous nearest Noun which replace with the Pronoun;

– lastly, replace Pronoun with previous nearest Noun.

Pseudocode 3: Noun-Pronoun Conversion

```
PROCEDURE conversionNounPronoun(article )
BEGIN PROCEDURE
    noun_position← find out Noun position of subject in sentence from article
    pronoun_position← find out Pronoun position of subject in sentence from article
    conversion_position← initialize empty list
    FOR i=1 to n
        replacing_pronoun← i th pronoun
        previous_noun← find out nearest previous noun from noun_position for replacing_pronoun
        conversion_position← add previous noun and replacing pronoun position
    ENDFOR
    FOR i=1 to n
        FOR j=1 to m
            article[i] ← replace i th word of article with conversion_position[j] if conversion_position[j] is the nearest noun of article[i]
    ENDFOR
    RETURN article
END PROCEDURE
```

Before splitting the text into sentences, pro-noun to noun conversion process has been added to the extractive based model. As a result, the problem has been solved which occurred after applying sentence ranking. After applying pronoun to noun con-version before splitting the text, the result has been seen in Fig. 9. The result shows that the modified extractive-based method maintains the sentence-to-sentence relationship.



Trump is a businessman and television personality who served as the 45th President of the United States from 2017 to 2021. Obama is a senior lecturer who forces on issues such as healthcare reform, climate change, and foreign policy. During the 2016 presidential election, Trump was the respective nominee of the Reputation. Trump won the election in a stunning upset, defeating Clinton in the electoral college despite losing the popular vote. Obama is also a well-known figure in American politics. Obama is better than him.

Fig. 9. After Applying Modified Extractive Based Method

2.4.2. Abstractive Based Method

The abstractive approach works based on the deep learning text summarization. This method generates new terms and phrases. It differs from the original article, but the generated terms and phrases are meaningful just like the same as the main article. The overall process of the abstractive-based model is shown in Fig. 10. In this model, three pre-trained models have used, such as google/pegasus-xsum, facebook/bart-large-cnn, and Yale-LILY/brio-cnndm-uncased. The google/pegasus-xsum model is trained with sampled gap sentences. The sampled gap sentence ratios on both HugeNews and C4. Second, the facebook/bart-large-cnn model is pre-trained by corrupting text with an arbitrary noising function. It learned to reconstruct the main article. The Yale-LILY/brio-cnndm-uncased model estimates the probability of system-generated summaries more accurately. All these models generate a great summary from the article. After generating the extraction summary using an extractive-based model, we pass that summary through the three pre-trained models.

The three models give us a number of three summaries, which are summary 1, summary 2 and summary 3. After generating these summaries, the final score was calculated for each summary using ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and with a reference summary. The maximum final score represents the great summary. After calculating the score, the maximum score was chosen. Finally, the best generated summary has been decided which final score is maximum among these. The whole working procedure has been stated in pseudocode 4.

Pseudocode 4: Abstractive Based Model

```
PROCEDURE generateAbstractiveSummary(reference)
BEGIN PROCEDURE
    summary_1 ← generate summary using google/pegus-xsum pretraind model
    summary_2 ← generate summary using facebook/bert-cnn-large pretraind model
    summary_3 ← generate summary using Yale-LILY/brio-cnndm-uncased pretrained model
    summary ← calculate final score each summary with reference and
    return maximum final score summary
    return summary
END PROCEDURE
```

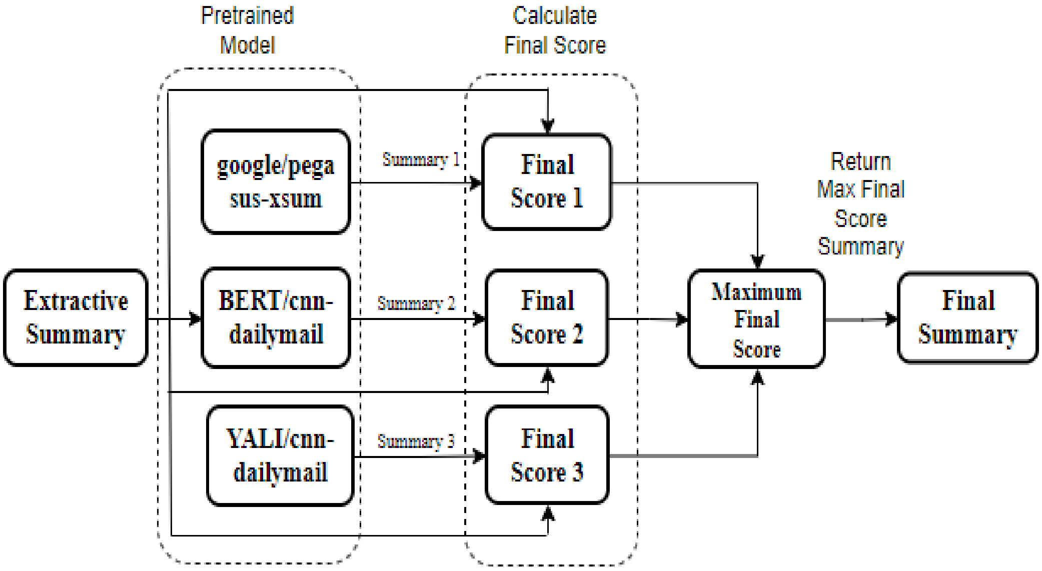


Fig. 10. Abstractive Based Model

3. Results and Discussion

The proposed text summarization model is compared with three state-of-the-art methods, including the abstraction summarization model (DEATS) [12], the joint entity and summary generation approach (JEANS) [35] and Plausibility-promoting Generative Adversarial Network (PGAN-ATSMT) [22]. The result has been experimented with the proposed model using cnn/dailymail datasets.

The summaries generated by the proposed system are also compared with corpus summaries using ROUGE metrics. ROUGE is a software package and set of metrics. It measures counting the overlapping unit numbers such as word sequence, word pairs, and n-gram between the candidate and reference summary. In the system experiments, each article has only one summary. In this system, the candidate or generated summary has been compared with a reference summary.

Table 1

Quantitate ROUGE Analysis for Proposed Model

No. of Document	Final Score		
	ROUGE-1	ROUGE-2	ROUGE-L
100	42.47	19.72	39.43
150	42.82	18.62	39.91
200	42.73	19.74	39.38
Average	**42.67**	**19.35**	**39.57**

Table 1 shows that the experimental results of the proposed model have been reported using a different number of articles on the cnn/daily-mail dataset. The proposed model achieves good results and the average results of ROUGE-1, ROUGE-2 and ROUGE-L are 42.67 %, 19.35 % and 39.57 % respectively. Fig. 11 has stated that the final score of the proposed model according to the different number of document experiments with that model. It has also been reported that the ROUGE score slightly differs with increasing number of documents. If the number of documents is 100, the ROUGE-1 score is 42.47 %. After increasing the number of documents from 100 to 200, the ROUGE-1 score increased by 0.25 %. On the other hand, increasing the number of documents from 100 to 150 decreased the ROUGE-2 score by 1.10 %. In table 2, the ground truth highlights and the generated summary.

The highlight summary manipulated who wrote the original article and the generated summary is produced by the proposed system. It has been shown that the summary generated using the proposed system is readable, meaningful and abstractive.

Fig. 12 represents the comparison between the existing and proposed models. The experiment results in Table 3 and Fig. 12 show that the system achieves better ROUGE scores for ROUGE-1 is 42.67 % and ROUGE-L is 39.57 %. It has also seen that the proposed model outperforms some previous methods such as JEANS (ROUGE-1 42.4 %, ROUGE-L 39.5 %), DEATS (ROUGE-1 40.85 %, ROUGE-L 37.13 %) and PGAN-ATSMT (ROUGE-1 42.15 %, ROUGE-L 38.94 %). In terms of ROUGE-1 and ROUGE-L metrics, the proposed model obtained the best performance. On the other hand, the existing model JEANS ROUGE-2 score is 20.2 %, which is better than the proposed model and some state-of-the-art models.

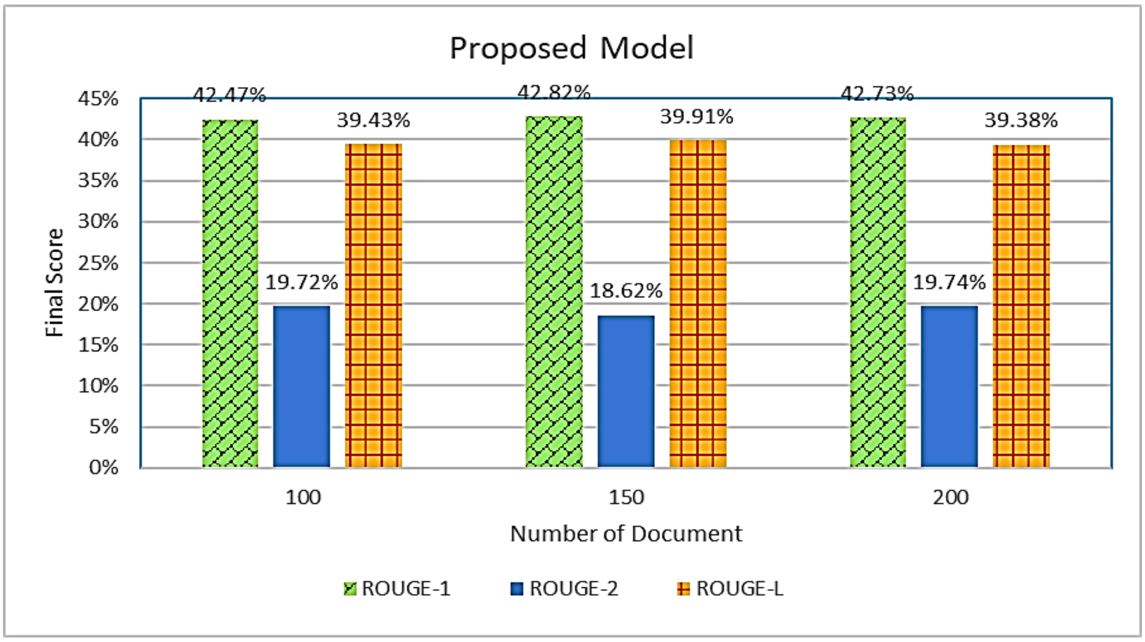


Fig. 11. Quantitate Final Score Analysis Different Number of Document

Table 2

The ground truth highlights and the generated summary using proposed model

Article: LONDON, England (Reuters) -- Harry Potter star Daniel Radcliffe gains access to a reported £20 million ($41.1 million) fortune as he turns 18 on Monday, but he insists the money won't cast a spell on him. Daniel Radcliffe as Harry Potter in "Harry Potter and the Order of the Phoenix" To the disappointment of gossip columnists around the world, the young actor says he has no plans to fritter his cash away on fast cars, drink and celebrity parties. "I don't plan to be one of those people who, as soon as they turn 18, suddenly buy themselves a massive sports car collection or something similar," he told an Australian interviewer earlier this month. "I don't think I'll be particularly extravagant. "The things I like buying are things that cost about 10 pounds -- books and CDs and DVDs." At 18, Radcliffe will be able to gamble in a casino, buy a drink in a pub or see the horror film "Hostel: Part II," currently six places below his number one movie on the UK box office chart. Details of how he'll mark his landmark birthday are under wraps…..(continue)
Highlights: "Harry Potter star Daniel Radcliffe gets £20M fortune as he turns 18 Monday. Young actor says he has no plans to fritter his cash away. Radcliffe's earnings from first five Potter films have been held in trust fund."
Summary by Proposed Model: "Harry Potter and the Order of the Phoenix" is breaking records on both sides of the Atlantic. The Londoner has filmed a TV movie called "My Boy Jack," about author Rudyard Kipling and his son." actor don't plan to be one of those people who, as soon as who turn 18, suddenly buy who a massive sports car collection or something similar," he said earlier this month.

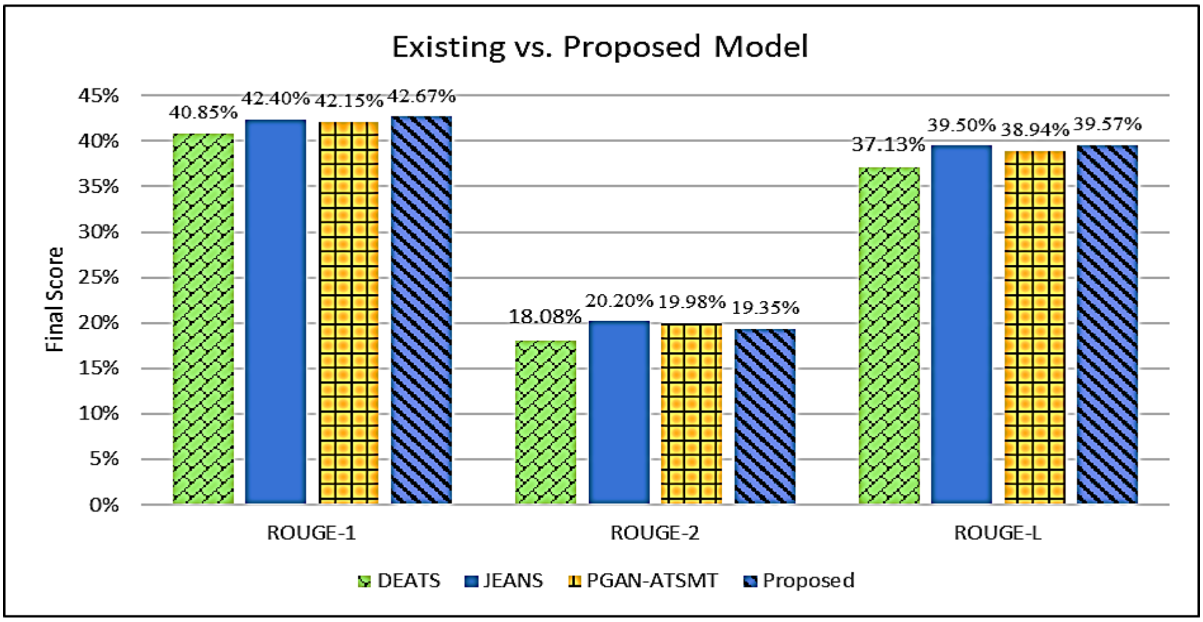


Fig. 12. Comparison Existing vs. Proposed Model

Table 3

Quantitate Final Score Analysis Existing Model with Proposed Model

Model	Final Score		
	ROUGE-1	ROUGE-2	ROUGE-L
JEANS	42.4	**20.2**	39.5
DEATS	40.85	18.08	37.13
PGAN-ATSMT	42.15	19.98	38.94
Proposed Model	**42.67**	19.35	**39.57**

Table 4

Quantitate Final Score Analysis and Differences with Proposed and Existing Model

Model	Final Score		
	ROUGE-1	ROUGE-2	ROUGE-L
DEATS vs. Proposed	+1.82 %	+1.27 %	+2.44 %
JEANS vs. Proposed	+0.27 %	-0.85 %	+0.07 %
PGAN-ATSMT vs. Proposed	+0.52 %	-0.63 %	+0.63 %

Comparison table 4 shows that the ROUGE-1 and ROUGE-L scores have increased in all comparisons. On the other hand, it also shows that ROUGE-2 score has increased only from DEATS model.

Conclusions

In this paper, an automatic text summarization model based on extractive and abstractive methods has been proposed.

1. In the extractive method, to extract some informative sentences using the sentence rank algorithm and overcome sentence-to-sentence relationship problems in sentence extraction has been proposed pronoun to noun conversion process.

2. In the abstractive method, the abstractive summary is generated using the three pre-trained models. These pre-trained models generated three abstractive summaries. The final summary has been decided to be based on the maximum ROUGE score of those summaries.

3. The model was tested on the cnn/daily-mail dataset. Experimental results showed that the model could generate readable, meaningful, and abstract summaries. In terms of ROUGE score, the model outperforms some state-of-the-art models for ROUGE-1 and ROUGE-L.

4. The proposed model does not achieve good results in ROUGE-2.

Future Research Development: Future work is to improve system efficiency and investigate the system to generate a more abstract and readable summary. Another goal of future work is to improve the ROUGE-2 score. Since the model tested only cnn/daily-mail dataset, for testing performance, we will experiment with the following datasets such as: Newsroom, Gigaword, Bigpatent etc.

Contributions of authors: coding analysis, comparison and drafting of the manuscript – **Md. Ahsan Habib**; research idea and led with overall supervision, revision, guidance, coding, analysis, comparison and drafting of the manuscript – **Romana Rahman Ema**; revision and guidance – **Tajul Islam**; result analysis, visualization, comparison and drafting of the manuscript – **Md. Yasir Arafat and Mahedi Hasan**.

All authors have read and agreed to the published version of the manuscript.

References

1. Opidi, Alfrick. *A Gentle Introduction to Text Summarization in Machine Learning*. Available at: https://blog.floydhub.com/gentle-introduction-to-text-summarization-in-machine-learning/ (accessed Oct. 28, 2022).

2. Awasthi, I., Gupta, K., Bhogal, P. S., Anand, S. S. and Soni, P. K. Natural language processing (NLP) based text summarization-a survey. In *2021 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India*, 2021, pp. 1310-1317. DOI: 10.1109/ICICT50816.2021.9358703.

3. Strilets, V., Donets, V., Ugryumov, M., Artiuch, S., Zelenskyi, R. and Goncharova, T. Agent-oriented data clustering for medical monitoring. *Radioelectronic and Computer Systems*, 2022, no. 1, pp. 103-114. DOI: 10.32620/reks.2022.1.08.

4. Anand, D. and Wagh, R. Effective deep learning approaches for summarization of legal texts. *Journal of King Saud University – Computer and Information Sciences*, 2022, vol. 34, no. 5, pp. 2141-2150. DOI: 10.1016/j.jksuci.2019.11.015.

5. Badi, H., Badi, I., El Moutaouakil, K., Khamjane, A. and Bahri, A. Sentiment analysis and prediction of polarity vaccines based on Twitter data using deep NLP techniques. *Radioelectronic and Computer Systems*, 2022, no. 4, pp. 19-29. DOI: 10.32620/reks.2022.4.02.

6. Villavicencio, P. and Watanabe, T. Text Summarization of Single Documents Based on Syntactic Sequences. In *Intelligent Interactive Multimedia Systems and Services. Smart Innovation, Systems and Technologies*, 2011, vol. 11, pp. 315-322. DOI: 10.1007/978-3-642-22158-3_31.

7. Keerthana, P. Automatic Text Summarization Using Deep Learning. *EPRA International Journal of Multidisciplinary Research (IJMR)*, 2021. Available at SSRN: https://ssrn.com/abstract=3837607. (accessed Oct. 28, 2022).

8. Chopra, S., Auli, M. and Rush, A. M. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, San Diego, California, 2016, pp. 93-98. DOI: 10.18653/V1/N16-1012.

9. Song, S., Huang, H. and Ruan, T. Abstractive text summarization using LSTM-CNN based deep learning. *Multimedia Tools and Applications*, 2019, vol. 78, pp. 857-875. DOI: 10.1007/S11042-018-5749-3.

10. Liu, L., Lu, Y., Yang, M., Qu, Q., Zhu, J. and Li, H. Generative adversarial network for abstractive text summarization. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018, vol. 32, no. 1, pp. 8109-8110. DOI: 10.1609/AAAI.V32I1.12141.

11. Barrera, A. and Verma, R. Combining syntax and semantics for automatic extractive single-document summarization. In *Computational Linguistics and Intelligent Text Processing. CICLing 2012. Lecture Notes in Computer Science*, 2012, vol. 7182, pp. 366-377. DOI: 10.1007/978-3-642-28601-8_31.

12. Khan, A., Salim, N., Farman, H., Khan, M., Jan, B., Ahmad, A., Ahmed, I. and Paul, A. Abstractive text summarization based on improved semantic graph approach. *International Journal of Parallel Programming*, 2018, vol. 46, pp. 992-1016. DOI: 10.1007/S10766-018-0560-3.

13. Li, P., Lam, W., Bing, L. and Wang, Z. Deep recurrent generative decoder for abstractive text summarization. *arXiv preprint arXiv:1708.00625*, 2017. DOI: 10.48550/arxiv.1708.00625.

14. Yao, K., Zhang, L., Du, D., Luo, T., Tao, L. and Wu, Y. Dual encoding for abstractive text summarization. *IEEE transactions on cybernetics*, 2020, vol. 50, no. 3, pp. 985-996. DOI: 10.1109/TCYB.2018.2876317.

15. Cai, T., Shen, M., Peng, H., Jiang, L. and Dai, Q. Improving transformer with sequential context representations for abstractive text summarization. In *Natural Language Processing and Chinese Computing. NLPCC 2019. Lecture Notes in Computer Science*, 2019, vol. 11838, pp. 512-524. DOI: 10.1007/978-3-030-32233-5_40.

16. Zhuge, H. 2 - The emerging structures. *In Computer Science Reviews and Trends, Multi-Dimensional Summarization in Cyber-Physical Society*. Morgan Kaufmann, 2016, pp. 21-43. DOI: 10.1016/B978-0-12-803455-2.00002-0.

17. Shi, L., Feng, S. K. and Zhu, Z. F. Functional hashing for compressing neural networks. *ArXiv Preprint, arXiv:1605.06560*, 2019, pp. 1-10. DOI: 10.48550/arXiv.1605.06560.

18. Roy, A. Understanding Automatic Text Summarization-1: Extractive Methods. Available at: https://towardsdatascience.com/understanding-automatic-text-summarization-1-extractive-methods-8eb512b21ecc (accessed Oct. 23, 2022).

19. Mihalcea, R. and Tarau, P. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404-411. Available at: https://aclanthology.org/W04-3252 (accessed: Oct. 23, 2022).

20. Krishnaveni, P. and Balasundaram, S. R. Generating fuzzy graph based multi-document summary of text-based learning materials. *Expert Systems with Applications*, 2023, vol. 214, article no. 119165. DOI: 10.1016/j.eswa.2022.119165.

21. Le, H. T. and Le, T. M. An approach to abstractive text summarization. In *2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR)*, Hanoi, Vietnam, 2013, pp. 371-376. DOI: 10.1109/SOCPAR.2013.7054161.

22. Rani, R. and Tandon, S. Literature review on automatic text summarization. *International Journal of Current Advanced Research*, 2018, vol. 7, iss. 2(C), pp. 9779-9783. Available at: https://journalijcar.org/issues/literature-review-automatic-text-summarization (accessed Nov. 08, 2022).

23. Alomari, A., Idris, N., Sabri, A. Q. M. and Alsmadi, I. Deep reinforcement and transfer learning for abstractive text summarization: A review. *Computer Speech & Language*, 2022, vol. 71, article no. 101276. DOI: 10.1016/J.CSL.2021.101276.

24. Zhang, J., Zhao, Y., Saleh, M. and Liu, P. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *37th International Conference on Machine Learning*, PMLR 119, 2020, pp. 11328-11339. DOI: 10.48550/arxiv.1912.08777.

25. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L. BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019. DOI: 10.48550/arxiv.1910.13461.

26. Yale LILY Lab. *Model Card for brio-cnndm-uncased*. Available at: https://huggingface.co/Yale-LILY/brio-cnndm-uncased (accessed Oct. 23, 2022).

27. Kamath, U., Liu, J. and Whitaker, J. *Deep learning for NLP and speech recognition*. Springer Cham, 2019. 621 p. DOI: 10.1007/978-3-030-14596-5.

28. *ROUGE (metric) - Wikipedia*. Available at: https://en.wikipedia.org/wiki/ROUGE_(metric) (accessed Oct. 23, 2022).

29. Lin, Chin-Yew. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, Barcelona, Spain. Association for Computational Linguistics, 2004, pp. 74-81. Available at: https://aclanthology.org/W04-1013. (accessed Oct. 23, 2022).

30. Sarwadnya, V. V. and Sonawane, S. S. Marathi extractive text summarizer using graph based model. In *2018 fourth international conference on computing communication control and automation (ICCUBEA)*, Pune, India, 2018, pp. 1-6. DOI: 10.1109/ICCUBEA.2018.8697741.

31. Horain, P., Achard, C. and Mallem, M. *Intelligent Human Computer Interaction: 9th International Conference, IHCI 2017, Evry, France, December 11-13, 2017, Proceedings*. Springer Nature, 2017. 216 p. DOI: 10.1007/978-3-319-72038-8.

32. Yavuz, S., Chiu, C. C., Nguyen, P. and Wu, Y. CaLcs: Continuously approximating longest common subsequence for sequence level optimization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3708-3718. DOI: 10.18653/V1/D18-1406.

33. Gowri, S. P. *CNN-DailyMail News Text Summarization*. Available at: https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail?resource=download (accessed Oct. 23, 2022).

34. Joshi, P. *An Introduction to Text Summarization using the TextRank Algorithm (with Python implementation)*. Available at: https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/ (accessed Nov. 13, 2022).

35. Nan, F., Nallapati, R., Wang, Z., Santos, C. N. D., Zhu, H., Zhang, D., McKeown, K. and Xiang, B. Entity-level factual consistency of abstractive text summarization. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021. pp. 2727-2733. DOI: 10.18653/V1/2021.EACL-MAIN.235.

Received 30.03.2023, Accepted 20.05.2023

АВТОМАТИЧНЕ РЕФЕРУВАННЯ ТЕКСТУ НА ОСНОВІ ЕКСТРАКТИВНО-РЕФЕРАТНОГО МЕТОДУ

Мухаммед Ахсан Хабіб, Руммана Рахман Емма*, Таджул Іслам, Мухаммед Ясір Арафат, Махеді Хасан

Вибір цього дослідження має значний вплив на повсякденне життя. У різних сферах, таких як журналістика, академічні кола, бізнес тощо, де великі обсяги тексту потрібно обробляти швидко й ефективно. Резюмування тексту - це техніка для створення точного та скороченого резюме просторих текстів. Створене резюме зберігає загальний зміст без втрати інформації та зосереджується на тих частинах, які містять корисну інформацію. Мета – розробити модель, яка перетворює велику статтю на стислі версії. Завдання, яке вирішується, – вибрати ефективну процедуру розробки моделі. Хоча нинішні моделі підсумовування тексту дають нам хороші результати в багатьох визнаних наборах даних, таких як cnn/daily-mail, newsroom тощо. Ці моделі не можуть вирішити всі проблеми. У цій статті запропоновано новий метод реферування тексту: комбінування техніки екстрактивного та абстрактного реферування тексту. У методі на основі вилучення модель генерує резюме за допомогою алгоритму ранжування речень і передає це згенероване резюме через абстрактний метод. Під час використання алгоритму ранжування речень після перегрупування речень зв'язок між реченнями руйнується. Щоб подолати цю ситуацію, у новій системі було запропоновано перетворення займенників в іменники. Після створення витягувального резюме, згенероване резюме було пропущене через абстрактний метод. Запропонована абстрактна модель складається з трьох попередньо підготовлених моделей: google/pegusus-xsum, face-book/bart-large-cnn model, Yale-LILY/brio-cnndm-uncased, яка генерує підсумкове резюме залежно від максимального кінцевого балу. Були отримані наступні результати: експериментальні результати на наборі даних CNN/daily-mail показують, що запропонована модель отримала оцінки ROUGE-1, ROUGE-2 і ROUGE-L відповідно 42,67%, 19,35% і 39,57%. Потім результат порівнювався з трьома найсучаснішими методами: JEANS, DEATS і PGAN-ATSMT. Результати перевершують найсучасніші моделі. Експериментальні результати також показують, що запропонована модель якісно більш читабельна та здатна генерувати абстрактні підсумки. Висновок: за показником ROUGE модель перевершує деякі сучасні моделі для ROUGE-1 і ROUGE-L, але не досягає хороших результатів у ROUGE-2.

Ключові слова: резюмування тексту; екстрактивне реферування; рефератне реферування; алгоритм ранжування речень; генерація тексту; конверсія іменників і займенників.

Мухаммед Ахсан Хабіб – магістр з інженерних наук вчений, каф. комп'ютерних наук та інженерії, Джашорський університет науки і технологій, Бангладеш.

Романа Рахман Ема (*відповідний автор) – PhD філософії каф. комп'ютерних наук та інженерії в Університеті інженерії та технології Кхулна, доц. каф. комп'ютерних наук та інженерії, Джашорський університет науки та технологій, Бангладеш.

Таджул Іслам – PhD філософії, каф. комп'ютерних наук та інженерії в Університеті інженерії та технології Кхулна, зав. каф. комп'ютерних наук та інженерії, Північно-Західний університет, Бангладеш.

Мухаммед Ясір Арафат – магістр з інженерних наук, доц. каф. комп'ютерних наук та інженерії, Джашорський університет науки та технологій, Бангладеш.

Махеді Хасан – магістр з технології баз даних, викладач інформатики та інженерії, Джашорський університет науки та технологій, Бангладеш.

Md. Ahsan Habib – M.Sc. Scholar in Engineering of Computer Science and Engineering Department, Jashore University of Science and Technology, Bangladesh,
e-mail: ahsan.just.cse@gmail.com, ORCID: 0000-0002-1119-4790.

Romana Rahman Ema (*corresponding Author) – PhD Scholar in Computer Science and Engineering Department at Khulna University of Engineering and Technology, Assistant professor of Computer Science and Engineering Department, Jashore University of Science and Technology, Bangladesh,
e-mail: rr.ema@just.edu.bd, ORCID: 0000-0002-2384-9539.

Tajul Islam – PhD Scholar in Computer Science and Engineering Department at Khulna University of Engineering and Technology, Assistant Professor, Head of Computer Science and Engineering Department, North Western University, Bangladesh,
e-mail: tajulkuet09@gmail.com, ORCID: 0009-0001-2504-6940.

Md. Yasir Arafat – M.Sc. in Engineering, Assistant Professor of Computer Science and Engineering Department, Jashore University of Science and Technology, Bangladesh,
e-mail: arafatcse@just.edu.bd, ORCID: 0000-0002-4278-147X.

Mahedi Hasan – M.Sc. in Database Technology, Lecturer of Computer Science and Engineering Department, Jashore University of Science and Technology, Bangladesh,
e-mail: m.hasan@just.edu.bd, ORCID: 0000-0002-7826-8600.