UDC 004.8
DOI https://doi.org/10.32782/2663-5941/2023.6/22

Sukhaniuk I.S.
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
https://orcid.org/0009-0000-4949-8549

Potapova K.R.
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
https://orcid.org/0000-0002-3347-6350

Nalyvaichuk M.V.
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
https://orcid.org/0000-0002-8942-9844

Vovk L.B.
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
https://orcid.org/0000-0002-3098-8078

TEXT SUMMARIZATION BASED ON TOPICRANK METHOD AND TEXT-TO-TEXT TRANSFORMER NEURAL NETWORK

The proposed text summarization system introduces a novel approach by combining the TopicRank method and the Text-to-Text transformer neural network to optimize the process of generating concise yet accurate summaries from large volumes of textual data. The primary goal of this research is to find an effective balance between processing speed and result accuracy in the context of handling extensive information datasets.

The problem addressed by this system lies in the complex interplay between the need for fast processing of large data volumes and the requirement for high precision in extracting information to create meaningful summaries. Both components of the system, namely TopicRank and the Text-to-Text transformer neural network, interact to achieve an optimal outcome.

Experiment results demonstrate the system's success in generating short summaries of large text documents within a limited time frame. This is achieved using the graph based TopicRank method to identify key sentences in the text. The obtained key sentences are then fed into the Text-to-Text transformer neural network, which, using deep learning, transforms them into informative summaries.

It is important to note that the system's performance depends on the quality of the input text and computational resources. Clean and structured input text yields better results, and high-performance computational resources enable faster processing of large data volumes. This underscores the importance of optimizing both input and computational processes to achieve optimal system performance.

The proposed system serves as an effective tool for text summarization in conditions involving the processing of large volumes of information. Its success in generating short and meaningful summaries indicates potential applications in areas where the speed of text processing and the preservation of a certain level of accuracy are crucial for obtaining meaningful information. Such an approach may find applications in fields where rapid analysis of extensive documentation is required, such as in scientific research, medical diagnostics, or intelligent information processing systems.

Key words: *natural language processing, text analysis with neural networks, transformer neural networks, TopicRank method, graph algorithm, text summarization.*

Formulation of the Problem. The main problem addressed in this work is the need to enhance the summarization process for large volumes of textual data. Contemporary information processing methods strive to strike a balance between processing speed and the accuracy of results, but challenges exist in resolving this issue.

One of the key difficulties is the necessity for processing speed when dealing with substantial amounts of text in situations where time is a critical factor. Specifically, in domains with massive information flows, such as scientific research or medical diagnostics, it is crucial to provide efficient and rapid text summarization methods. However, it

is important to avoid compromising the accuracy and relevance of the summarized information.

Another problem is the selection of an optimal method to achieve this balance. In the context of this work, two primary components are employed: the TopicRank method and the Text-to-Text transformer neural network. Addressing how these two approaches can interact optimally to ensure effective summarization is a crucial aspect explored in this work.

Therefore, the problem statement in this research is defined by the need for an efficient summarization system that ensures the processing speed of large volumes of data without sacrificing the accuracy and relevance of the summarized information. Finding the optimal balance between these aspects in the context of combining the TopicRank method and the Text-to-Text transformer neural network becomes a key task that defines the direction of the research.

Analysis of recent research and publications. This section discusses three key architectures: Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN, and Transformers. Each of these architectures is used for summarizing textual information, but they differ in their structure and approach to processing sequential data.

Analyzing these architectures in the context of automatic text summarization will help identify their advantages and limitations, as well as determine the optimal choice for specific tasks.

1. *Using CNN for text summarization*:

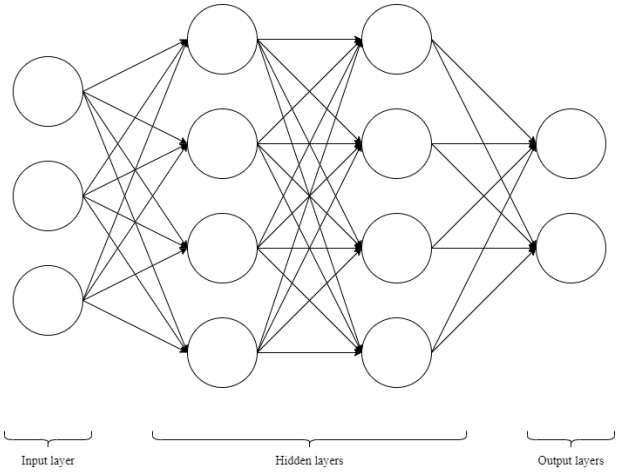


Fig. 1. Schematic representation of CNN

In this approach, a deep learning network known as Convolutional Neural Network (CNN) is utilized for the automatic summarization of textual information. The fundamental idea is to use convolutional layers to extract key features from the text, followed by fully connected layers for generating the actual summary.

Research [6] in this direction includes the LSTM-CNN model, which implements abstractive text summarization. This model employs CNN to extract "semantic phrases" from sentences. Following this, it uses a Long Short-Term Memory (LSTM) network to create the final summary of the text. The particularity of this method lies in its simplicity and efficiency. However, it is worth noting that it may not consider the context and nuances of the original text.

Therefore, the use of CNN for text summarization becomes an attractive strategy, especially when the semantic aspect of the text is crucial. Still, its limitations in understanding a broader context or details of the text should be considered [2, c. 233].

2. *Using RNN for text summarization*:

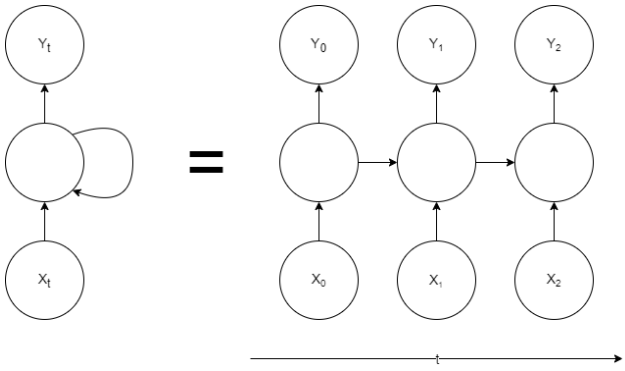


Fig. 2. Schematic representation of RNN

This approach to automatic text summarization is based on the use of Recurrent Neural Networks (RNN). The fundamental concept involves modeling sequences of words in the text to effectively interact with the context and create a compact representation of the text.

In the examined research [7], an Encoder-Decoder RNN model was introduced for the task of text summarization. In this model, the RNN acts as an encoder responsible for transforming the input text into a compact representation that retains key information. Additionally, using another RNN as a decoder, the summary is generated.

This approach emphasizes sequential analysis of the text, allowing for the creation of more comprehensive summaries by considering the relationships between words. However, it is important to note that RNN may encounter the issue of "vanishing gradients" with lengthy text, which can affect the quality of the summaries [3, c. 67].

3. *Using transformer neural network for text summarization*:

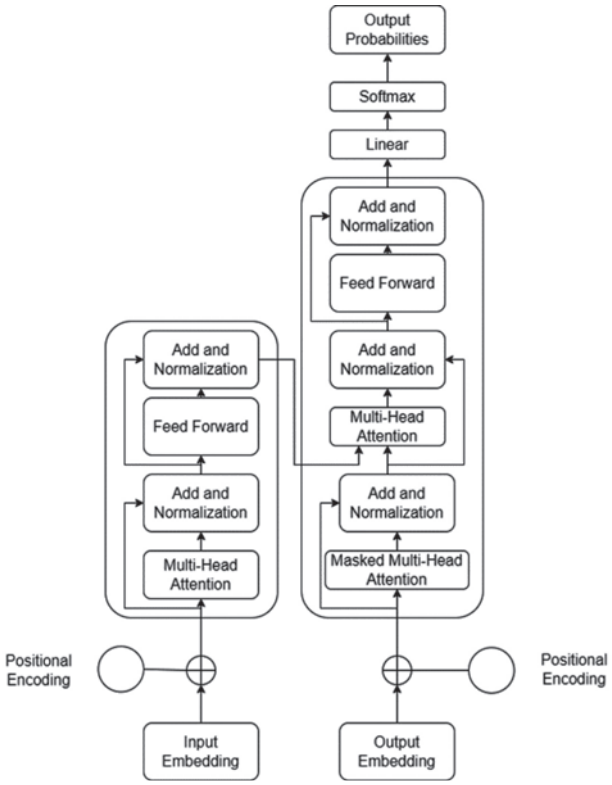


Fig. 3. Schematic representation of the transformer neural network model

This approach to automatic text summarization employs transformers. The main idea is that transformers use attention mechanisms to model the context of words, sentences, and even entire documents. This enables them to effectively consider interactions and the importance of different parts of the text when generating a summary.

In the discussed research [5], the Text-to-Text Transfer Transformer (T5) model is mentioned, which is used for abstractive text summarization. This model has been trained on a large amount of textual data, allowing it to deeply understand language patterns and relationships. Transformers are capable of highly efficient modeling of sequences and context, and their attention mechanisms enable effective handling of text summarization tasks.

This approach is particularly important when working with large volumes of textual data, as transformers can detect deep dependencies and complex relationships in the text. Regarding limitations, it's important to note that the use of transformers may require significant computational resources [1, c. 10].

4. *Comparison of architectures*

Each architecture has its strengths and weaknesses. CNNs excel in processing structured data such as images but may overlook the context and nuances of text. RNNs are proficient in tasks related to sequences and can model relationships between words, yet they encounter the issue of "vanishing gradients" in long texts. Transformers, particularly T5, exhibit impressive versatility and the ability to model contextual information in text but require significant computational resources.

Comparing these architectures, the choice of a specific one depends on the task at hand and the nature of the input data. CNNs may prove effective for processing structured data, RNNs for sequences, and transformers for tasks where context and a large volume of training data are crucial [2, c. 238; 3, c. 71].

The purpose of the article. The aim of the research is to develop a text summarization system that combines the TopicRank method [4] and the Text-to-Text transformer neural network [5].

Presenting main material

1. TopicRank method

The main idea of the method is to use a graph structure to analyze and highlight key topics from the text. This process can be clearly illustrated in the flowchart below.

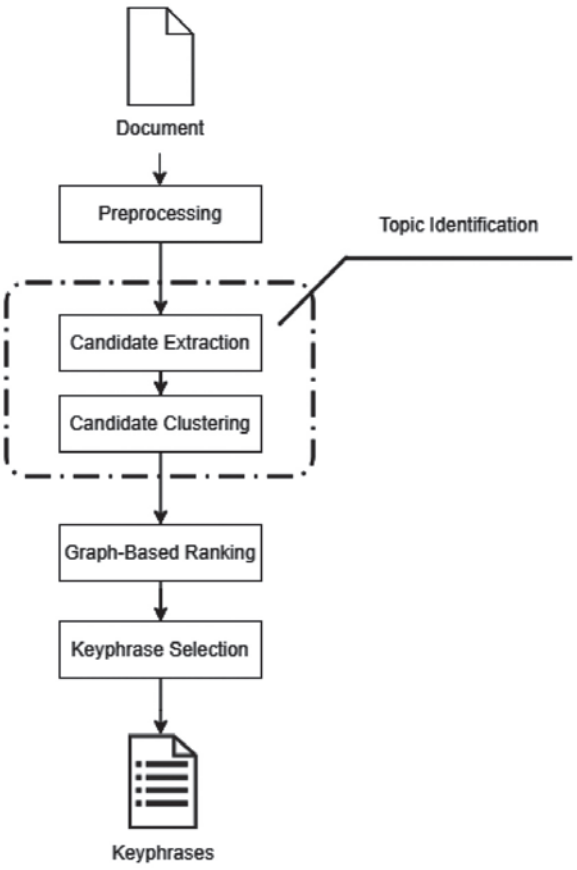


Fig. 4. Schematic representation of the stages of TopicRank method

1.1. Topic identification

The first stage in the TopicRank method is the identification of key themes and phrases that represent them. To achieve this, an method for identifying candidate key phrases is employed, aiming to best reflect the document's topics. Method follows the approach proposed by Wan and Xiao (2008), selecting the longest sequences of nouns and adjectives from the document as candidate key phrases. Other methods may utilize syntactically filtered n-grams containing the highest number of candidates corresponding to references to key phrases. However, the limited length of n-grams can be problematic, as they may not always capture as much information as the longest sequences of nouns. Additionally, they are less likely to correspond to grammatically correct text.

1.2. Graph-Based Ranking

The TopicRank method represents a document as a complete graph, where vertices correspond to topics, and edges have weights based on the strength of semantic connections between vertices. Subsequently, a graph-based ranking model, TextRank, is employed to assign significance to each topic.

1.3. Graph construction

Formally defining, let $G = (V, E)$ be a complete and undirected graph, where V is the set of vertices, and E is a subset of $V \times V$ The vertices correspond to topics, and the edge between two topics t_i and t_j has a weight determined by the strength of the semantic connection between the vertices. The weight $w_{i,j}$ of the edge between t_i and t_j is calculated as follows:

$$w_{i,j} = \sum_{c_i \in t_i} \sum_{c_j \in t_j} dist\left(c_i, c_j\right) \qquad (1)$$

The inverse distances $dist\left(c_i, c_j\right)$ between the positions of keyword candidates c_i and c_j in the document are computed by the formula:

$$dist\left(c_i, c_j\right) = \sum_{p_i \in pos(c_i)} \sum_{p_j \in pos(c_j)} \frac{1}{\left|p_i - p_j\right|} \qquad (2)$$

Here, $dist\left(c_i, c_j\right)$ denotes the inverse distances between the positions of keyword candidates c_i and c_j, and $pos\left(c_i\right)$ includes all positions of keyword candidates c_i.

The approach to constructing the graph differs from the TextRank method. Graph *G* is a complete graph, leading to interconnected topics. The completeness of the graph has an advantage over exploring relationships between topics. Additionally, computing weights based on distances avoids the need for manual parameter definition, such as the window size used in recent methods (TextRank, SingleRank, etc.)

1.4. Keyphrases selection

The final step in TopicRank is the selection of a single representative from the key phrase candidates for each topic. This selection helps avoid duplication and ensures comprehensive coverage of the document's topics.

To choose the candidate that best represents the topic, three strategies are proposed. The first strategy involves selecting the key phrase candidate that appears first in the document. The second strategy assumes that the most commonly used form of the topic is the most representative and selects the key phrase candidate that is used most frequently. The third strategy chooses the centroid of the cluster. The centroid is a key phrase candidate that is most similar to other candidates in the cluster.

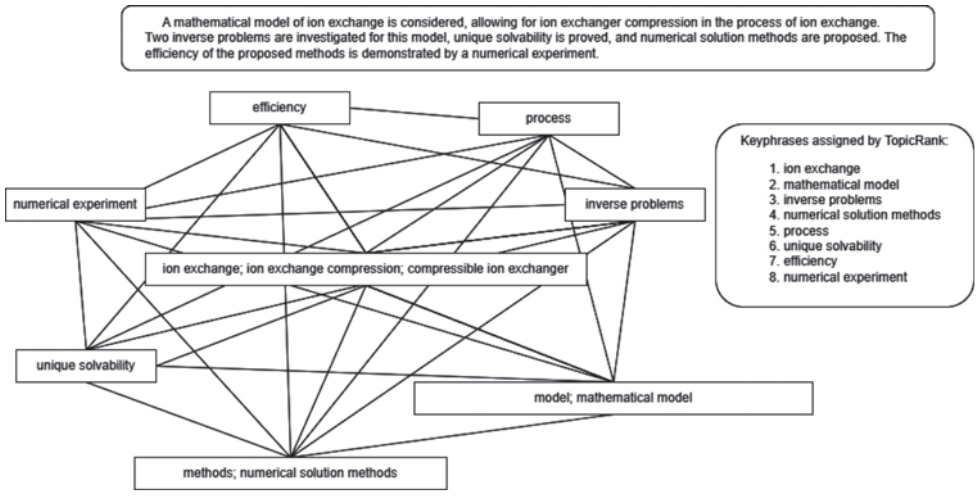


Fig. 5. Sample graph build by TopicRank [4]

```
def LoadT5Tokenizer(self):
    self.logger.info("Loading T5Tokenizer")
    self.tokenizer = T5Tokenizer.from_pretrained('t5-base', model_max_length=MAX_LENGTH)
    self.logger.info("Loaded T5Tokenizer")

def LoadT5ForConditionalGeneration(self):
    self.logger.info("Loading T5ForConditionalGeneration")
    self.model = T5ForConditionalGeneration.from_pretrained('t5-base')
    self.logger.info("Loaded T5ForConditionalGeneration")
```

Fig. 6. Initialization of T5 model and tokenizer

```
def GetSummarizedText(self, input_article: str, length_penalty: float, num_beams: int) -> str:
    inputs = self.tokenizer.encode(
        "summarize: " + input_article,
        return_tensors="pt",
        max_length=MAX_LENGTH,
        truncation=True
    )
    outputs = self.model.generate(
        inputs,
        max_length=MAX_LENGTH,
        min_length=16,
        length_penalty=length_penalty,
        num_beams=num_beams,
        early_stopping=True
    )
    return self.tokenizer.decode(outputs[0], skip_special_tokens=True)
```

Fig. 7. Usage of T5 model and tokenizer

2. Нейронна мережа Text-To-Text (T5)

The second key component of our method is the Text-To-Text (T5) neural network from Hugging Face. The uniqueness of T5 lies in its ability for abstractive text summarization, as well as in its transformation of all tasks into a text-to-text format.

2.1. Text Preprocessing

Text preprocessing is a key stage in addressing the Text-To-Text task using transformer neural networks such as T5. At this stage, the text undergoes a comprehensive set of operations to prepare it for further processing by the model.

– *Tokenization*: The first step is tokenization, which involves dividing the text into individual tokens (e.g., words or subwords). This is necessary for representing the text as a sequence of tokens, simplifying its handling for the model.

– *Cleaning and normalization*: After tokenization, text cleaning and normalization are performed. Cleaning includes removing unnecessary characters, such as punctuation and numbers, which could introduce noise into the model. Normalization may involve transforming the entire text to lowercase for uniformity and avoiding unnecessary variations in form.

– *Stop-word removal*: For effective representation of the text in the model, stop words, such as "and," "the," "of," which often carry little significant information, may be removed, reducing data volume.

– *Vectorization*: In the final stage of preprocessing, the input text is transformed into a numerical form that can be understood by the model. This may include using vectorization techniques like Word Embeddings to obtain numerical embeddings of tokens.

The overall goal of preprocessing is to simplify and optimize the representation of the input text for further processing by the T5 model, ensuring the separation of essential informational elements from unnecessary noise and providing clear data presentation for the model.

2.2. Encoding

The encoding stage plays a crucial role in preparing the input text for further processing by a transformer neural network, such as T5. This stage involves transforming the text or sequence of tokens

into vector representations that can be understood and processed by the model.

– *Token Embedding*: The central element of encoding is the creation of embeddings for each token in the text. Embeddings are vector representations of tokens that convey the semantic meaning of words. A pretrained model, such as BERT or T5, which has embedded knowledge of the semantics of words based on a broad corpus of textual data, can be used for this purpose.

– *Positional Encoding*: In addition to embeddings, the model can use information about the position of tokens in the text. Since transformers do not consider the order of words in the input sequence, positional information is added to determine where the tokens are located in the text.

Encoding creates an input matrix where each row represents an embedding for a specific token and includes additional vectors to account for the positions of tokens. This stage provides a temporary and spatial representation of the input text, which can be presented to a transformer neural network for further processing and generation of the output text.

2.3. Decoding

Decoding is a crucial stage in addressing Text-To-Text tasks, as it is during this stage that the model generates the output text based on the encoded vector representation of the input text.

– *Autoregression*: At the beginning of decoding, the model uses prepared embeddings and information about encoded tokens to generate the first token of the output text. After generating the first token of the output text, the obtained result is used as partial input for generating the next token, and so on. This process is known as autoregression, where each new token is generated based on the previous one.

– *Attention Mechanisms*: During decoding, attention mechanisms are employed to consider the context and dependencies between tokens in the input text. This helps the model focus on important parts of the text while generating each token of the output text. Attention mechanisms can be implemented in various ways, such as the Transformer attention mechanism.

Decoding is completed when the model generates a token indicating the end or maximum length of the output text. The result obtained is the generated output text, which can be used to address a specific task, such as translation or answering a question. Decoding requires careful management of text generation to ensure logical, grammatical, and contextually relevant output text.

2.4. Post-processing

Post-processing is an important final stage in working with the results generated by the neural network in the preceding Text-To-Text processing stages. This stage is designed to enhance the readability, correctness, and logical coherence of the output text.

– *Error Correction*: One of the main aspects of post-processing is error correction. Neural networks, including transformers, may occasionally generate incorrect or ungrammatical text. Error correction mechanisms may involve automatic corrections of grammatical or structural mistakes.

– *Removal of Redundant Code*: During text generation, the model may add unnecessary tokens or symbols that do not provide additional information and may impact the clarity of the text. The post-processing stage may involve removing such redundant elements from the output text.

– *Optimization of Text Structure*: The model may generate text that is not perfectly organized or does not adhere to acceptable structural conventions. Post-processing may include restructuring or optimizing the text to achieve greater clarity and logical coherence.

– *Refinement of Style*: Models may also have limitations in understanding specific speech styles, tone, or terminology. Post-processing may involve refining the stylistic aspects of the text to align with a particular context or audience expectations.

– *Logical Consistency Check*: The final aspect of post-processing is checking the logical consistency and coherence of the text. This may involve analyzing the interaction between different parts of the text and correcting any illogical statements.

Post-processing aims to improve the generated text by ensuring its relevance to the task and maximizing its clarity and acceptability for the end user.

3. Combination of methods

The combination of the TopicRank method and the T5 neural network ensures greater efficiency in text summarization. The TopicRank method helps identify key phrases and important sentences, while T5 uses this information to generate an abstractive summary that incorporates the main themes and ideas of the original text. This combination enables the neural network to focus on the key text, reducing the time required to create summaries.

Thus, employing the TopicRank method to extract key elements of the text and subsequently utilizing them in the T5 neural network provides enhanced efficiency in the task of text summarization. This approach not only automates the summarization process but also ensures the preservation of accuracy and relevance in summaries while reducing the time required for their creation. Such an approach can be beneficial in various fields where the processing of large amounts of textual information is crucial.

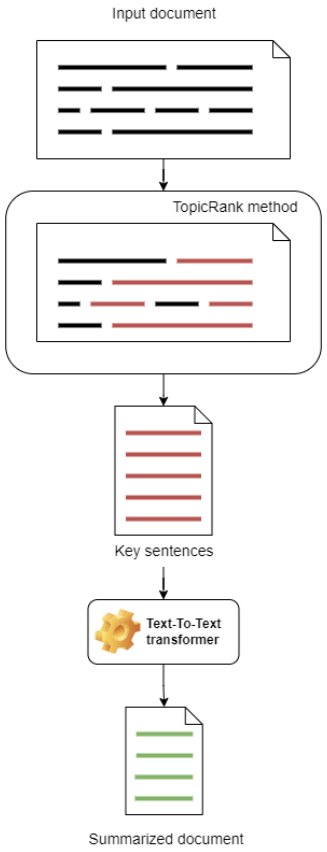
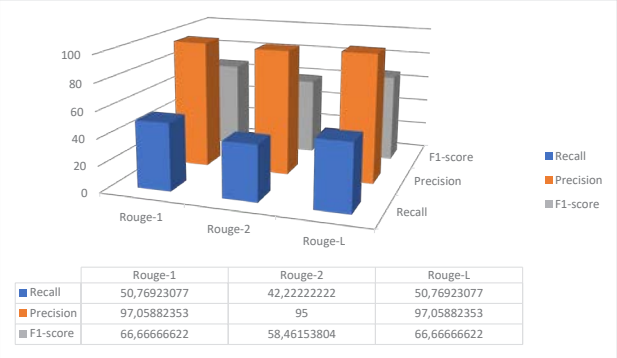


Fig. 8. Combination of methods

4. Results

Without using compression using the TopicRank method. Execution time: 46.831591844558716 seconds.



	Rouge-1	Rouge-2	Rouge-L
Recall	50,76923077	42,22222222	50,76923077
Precision	97,05882353	95	97,05882353
F1-score	66,66666622	58,46153804	66,66666622

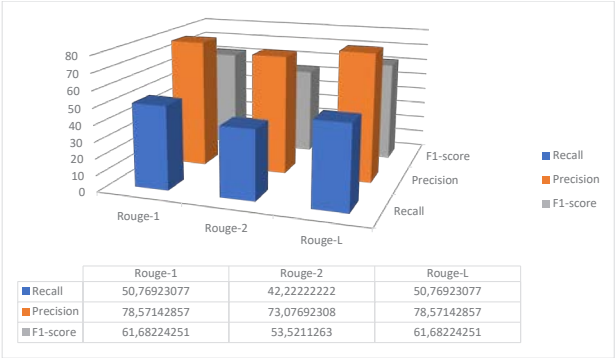
Fig. 9. Rouge metric without using compression using the TopicRank method

Using 40 keywords compression. Execution time: 38.91822385787964 seconds.

Conclusion. The research conducted aimed to find an optimal balance between time efficiency and accuracy in text summarization, employing a combination of the TopicRank method and the T5 neural network. Without TopicRank compression, method executed in 46.83 seconds. Introducing 40 keywords for compression reduced the execution time to 38.92 seconds, and further compression to 20 keywords resulted in an execution time of 33.35 seconds.

The ROUGE metric values revealed a trade-off between time optimization and result accuracy. The use of 40 keywords for compression led to a decrease in ROUGE-1, ROUGE-2, and ROUGE-L compared to the uncompressed variant. Moreover, the use of 20 keywords further exacerbated this reduction in metrics.

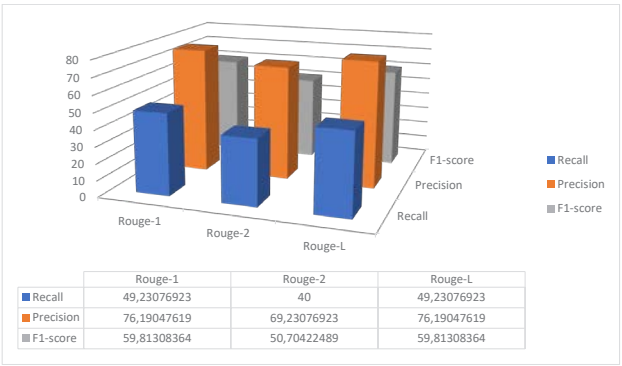
Despite the decrease in accuracy in percentage terms, the integration of the TopicRank method for extracting key elements from the text, followed by their utilization in the T5 neural network, demonstrated significant efficiency in text summarization. The heightened speed in text processing can be pivotal in handling large datasets where system speed and responsiveness are of paramount importance.



	Rouge-1	Rouge-2	Rouge-L
Recall	50,76923077	42,22222222	50,76923077
Precision	78,57142857	73,07692308	78,57142857
F1-score	61,68224251	53,5211263	61,68224251

Fig. 10. Rouge metric Using 40 keywords compression

Using 20 keywords compression. Execution time: 33.35249924659729 seconds.



	Rouge-1	Rouge-2	Rouge-L
Recall	49,23076923	40	49,23076923
Precision	76,19047619	69,23076923	76,19047619
F1-score	59,81308364	50,70422489	59,81308364

Fig. 11. Rouge metric Using 20 keywords compression

References:

1. Суханюк І. С., Івасенко Д. В., Потапова К. Р. Використання нейронних мереж для аналізу текстів // Міжнародний науковий журнал «Інтернаука». 2023. № 13. https://doi.org/10.25313/2520-2057-2023-13-9036
2. Sukhaniuk I. S., Potapova K. R. USAGE OF CONVOLUTIONAL NEURAL NETWORKS IN NATURAL LANGUAGE PROCESSING. p. 233. URL: https://sci-conf.com.ua/wp-content/uploads/2023/09/SCIENTIFIC-RESEARCH-IN-THE-MODERN-WORLD-21-23.09.23.pdf
3. Sukhaniuk I. S., Potapova K. R. USAGE OF RECURRENT NEURAL NETWORKS IN NATURAL LANGUAGE PROCESSING. p. 67. URL: https://sci-conf.com.ua/wp-content/uploads/2023/09/MODERN-RESEARCH-IN-SCIENCE-AND-EDUCATION-14-16.09.23.pdf
4. Adrien Bougouin, Florian Boudin, Beatrice Daille. TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction. URL: https://aclanthology.org/I13-1062.pdf
5. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. URL: https://arxiv.org/pdf/1910.10683v4.pdf
6. Shengli Song, Haitao Huang, Tongxiao Ruan. Abstractive text summarization using LSTM-CNN based deep learning. URL: https://sci-hub.se/10.1007/s11042-018-5749-3
7. Shaik Rafi, Ranjita Das. RNN Encoder And Decoder With Teacher Forcing Attention Mechanism for Abstractive Summarization. URL: https://doi.org/10.1109/INDICON52576.2021.9691681

Суханюк І.С., Потапова К.Р., Наливайчук М.В., Вовк Л.Б. УЗАГАЛЬНЕННЯ ТЕКСТУ НА ОСНОВІ МЕТОДУ TOPICRANK ТА TEXT-TO-TEXT ТРАНСФОРМЕРНОЇ НЕЙРОННОЇ МЕРЕЖІ

Запропонована система підсумовування тексту представляє новий підхід, поєднуючи метод TopicRank та Text-to-Text трансформерну нейронну мережу для оптимізації процесу генерації коротких, але точних підсумків з великих обсягів текстових даних. Головною метою цього дослідження є пошук балансу між швидкістю виконання та точністю результатів у контексті обробки об'ємних інформаційних наборів.

Проблема, яку вирішує ця система, полягає у комплексній взаємодії між необхідністю швидкої обробки великих обсягів даних і необхідністю точності виділення інформації для створення змістовних підсумків. Обидва компоненти системи, а саме TopicRank і Text-to-Text трансформерна нейронна мережа, взаємодіють для досягнення оптимального результату.

Результати експериментів свідчать про ефективність системи у генерації коротких підсумків великих текстових документів в обмежений часовий проміжок. Це досягається завдяки використанню графового алгоритму TopicRank для виділення ключових речень у тексті. Отримані ключові речення передаються Text-to-Text трансформерній нейронній мережі, яка, використовуючи глибоке навчання, перетворює їх в інформативні підсумки.

Важливо відзначити, що продуктивність цієї системи залежить від якості вхідного тексту та обчислювальних ресурсів. Чистий і структурований вхідний текст забезпечує кращі результати, а високопродуктивні обчислювальні ресурси дозволяють швидше обробляти великі обсяги даних. Це підкреслює важливість оптимізації як введення, так і обчислювального процесу для досягнення оптимальної продуктивності системи.

Запропонована система являється дієвим інструментом для підсумовування тексту в умовах обробки великих обсягів інформації. Її позитивний результат в генерації коротких і змістовних підсумків вказує на потенційне застосування в сферах, де важлива швидкість обробки тексту та збереження певної точності для отримання значущої інформації. Такий підхід може знайти застосування в областях, де потрібно швидко аналізувати великі обсяги документації, наприклад, у наукових дослідженнях, медицинській діагностиці або інтелектуальних системах обробки інформації.

Ключові слова: *обробка природної мови, аналіз тексту за допомогою нейронних мереж, трансформерні нейронні мережі, метод TopicRank, графовий алгоритм, узагальнення тексту.*