Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 12, No. 3, 2024, 325-335.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

PTRP: Title Generation Based on Transformer Models

Davud Mohammadpur* and Mehdi Nazari

Computer Department, University of Zanjan, Zanjan, Iran.

Article Info

Article History:
Submit Date 18 June 2024
Accept Date 05 September 2024
Accepted 18 October 2024

DOI:10.22044/jadm.2024.14633.2565

Keywords:
Text Summarization, TextRank, Title Generation, Transformer, BART, T5, PEGASUS, Rouge Measure.

**Corresponding author: dmp@znu.ac.ir (D. Mohammadpour).*

Abstract

Text summarization has become one of the favorite subjects of researchers due to the rapid growth of contents. In title generation, a key aspect of text summarization, creating a concise and meaningful title is essential as it reflects the article's content, objectives, methodologies, and findings. Thus, generating an effective title requires a thorough understanding of the article. Various methods have been proposed in text summarization to automatically generate titles, utilizing machine learning and deep learning techniques to improve results. This study aims to develop a title generation system for scientific articles using transformer-based methods to create suitable titles from article abstracts. Pre-trained transformer-based models like BERT, T5, and PEGASUS are optimized for constructing complete sentences, but their ability to generate scientific titles is limited. We have attempted to improve this limitation by presenting a proposed method that combines different models along with a suitable dataset for training. To create our desired dataset, we collected abstracts and titles of articles published on the ScienceDirect.com website. After performing preprocessing on this data, we developed a suitable dataset consisting of 50,000 articles. The results from the evaluations of the proposed method indicate approximately 4% improvement based on various ROUGE metrics in the generation of scientific titles. Additionally, an examination of the results by experts in each scientific field revealed that the generated titles are also acceptable to these specialists.

1. Introduction

With the significant growth of online information and documents, the volume of textual data has increased exponentially, bringing significant challenges to tasks such as document management, text classification, and information retrieval. Automatic text summarization (ATS) is becoming an important tool to address these challenges [1]. The summary is defined as follows: "a text that is produced from one or more texts that convey important information in the original text, and that is no longer than half of the original text(s) and usually significantly less than that" [2]. Automatic text summarization is an application that has various uses. One of these applications is title generation for scientific articles, which can automatically suggest a suitable title for an article based on its abstract [2, 3]. Automatic text summarization is the production of concise and fluent summaries while preserving key information content and overall meaning. It has many complexities because when we summarize a text, we read it entirely to increase our understanding and then write a summary highlighting its main points. Since machines need more human knowledge and language ability, automatic text summarization becomes complicated and challenging [2, 4]. The title of an article is a summarization of the article, and determining a compelling title for the article is very important [5]. This research aims to produce a suitable title for scientific articles. For this purpose, we propose a method to produce titles from the abstract of scientific articles using summarization methods based on deep learning and transformers.

Transformer models, as deep neural models are considered suitable for natural language generation. Due to their success in summarizing documents [6], we also use transformer models to present our proposed method. Using these models requires a suitable amount of data as learning data. However, the absence of a rich, orderly, and regular data set that includes articles from various fields of science is challenging. On the other hand, preparing a suitable dataset is also very time-consuming and sometimes associated with limitations. Also, due to the heavy and bulky nature of training models and the creation of many parameters during model training, the required processes require very powerful hardware resources and are very time-consuming.

The nature of the title of an article is different from the summary of a text. For example, the summary of a text should consist of complete and concise sentences, but a title does not have to be in the form of a sentence, mainly in the form of a pseudo-sentence, and it should consist of several words that include essential terms. Also, sometimes abbreviations are used in the title of articles, which do not exist in the dictionary of words, so such cases should be considered in the production of the title. For this reason, unlike other research, we have avoided removing abbreviations, stop words, and numbers in preprocessing.

This article is written in four parts. The first part discussed the concepts, problems, challenges, and importance of the subject. In the subsequent section, we shall examine the transformer architecture and its diverse methodologies, particularly in the domains of natural language processing, summarization, and text generation, as well as examine related literature. In the third part of the presentation, we present our dataset, the proposed method, and its architecture for conducting the research. In the fourth part, we will report the results and evaluations.

2. Background

Deep Neural Networks (DNN) are an essential infrastructure and advanced solution for most learning-based text-processing tasks. However, most common neural network techniques need to retain the true meaning of the context [4, 7]. Transformers are a type of deep neural network that solves this problem. Transformers use a "multi-head self-attention" mechanism to extract features. Unlike conventional DNN methods, transformers use the attention mechanism to learn a complete part of a sequence with the help of encoding and decoding blocks. One of the critical advantages of transformers over conventional DNN methods is their ability to truly understand the context due to their attention mechanisms.

2.1. Transformer model architecture

Transformers architecture may change for different applications based on their specific needs. The basic architecture of transformers is developed based on the autoregressive sequence transformation model, which includes two main modules: Encoder and Decoder. These modules are executed several times depending on the task requirements. Encoder and Decoder modules contain several layers. Attentional mechanisms are also used in the general architecture of the transformer model for text-processing tasks, as shown in Figure 1. The attention mechanism is implemented several times in parallel in the transformer architecture, which is why several "Heads" exist in this architecture [8].

2.1.1. Encoder module

The stack module in the transformer architecture consists of two primary layers: the "Feed Forward" and the “Multi-Head Attention”. In addition, it contains the remaining connections around both layers and two layers of Add & Norm, which play a critical role [8]. In the case of text processing, the Encoder module receives an embedded input that is created based on the meaning and position information of the input through the Embedding and Position Encoding layers. From the embedded input, three matrices (Query(Q), Key(K), and Value(V)) are generated along with positional information that passes through the "Multi-Head Attention" layer [9]. The Feed Forward layer addresses the issue of rank degradation that can occur in the computation process. In addition, a normalization layer is applied to each step, which reduces the dependency between layers by normalizing the weights used in the gradient calculation in each layer. As shown in Figure 1, to address the issue of Vanishing Gradient, Residual Connection is applied to every output of both the attention and Feed-Forward layers [9].

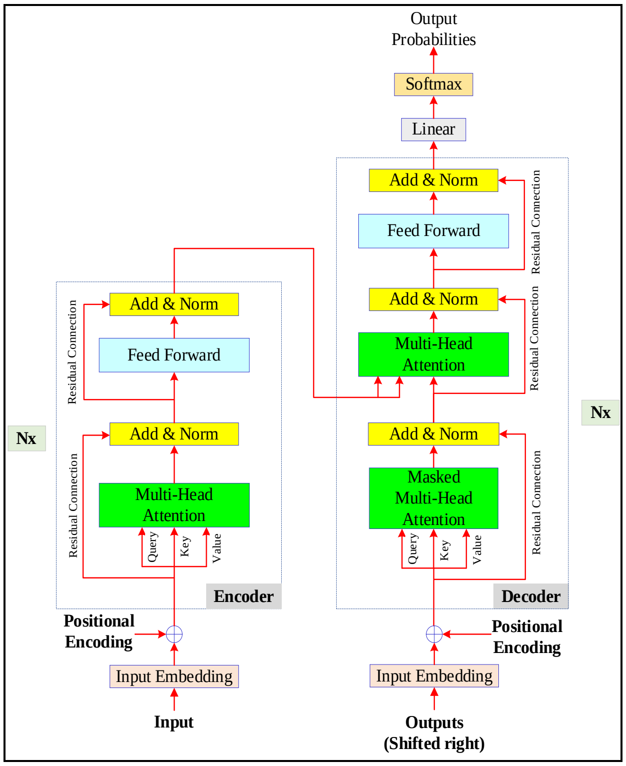

Figure 1. Transformer architecture [8].

2.1.2. Decoder module

The Decoder module in the transformer architecture is similar to the Encoder module, which, in addition to Feed-Forward, multi-head Attention, Residual Connection, and Add & Norm layers, also has "Masked Multi-Head Attention" layers. These layers use the Scaled Dot Product and Mask operations to exclude future predictions and consider only previous outputs [9].

The attention mechanism is applied twice in the Decoder: one for computing Attention between elements of the targeted output and another for finding Attention between the encoding inputs and targeted output. Each attention vector is then passed through the Feed-Forward unit to make the output more comprehensible for the layers. Next, the decoding result produced by the Linear and SoftMax layers is obtained on top of the Decoder to calculate the final output of the transformer architecture.[8].

2.2. Transformer models for summarization

Article title generation is in the context of natural language processing and the subcategory of text summarization. Therefore, in this section, we will review the common transformer models in text summarization. Figure 2 shows the commonly used transformer models in the summarization. Various models of transformers have been proposed in summarizing the text, and some of the models that have had more successful results include the following:

- ***PEGASUS:*** This model is a good text summarization model that uses a transformer Encoder and Decoder modules [10]. While models based on masked language modelling cover only a tiny part of the text, PEGASUS hides the entire multiple sentences, selects the masked sentences based on their importance, and produces them as output. This model has shown remarkable performance on unknown summarization datasets.
- ***T5:*** The T5 model is based on the transformer architecture and the text-to-text approach. T5's ability to capture hierarchical representations, manage long-range dependencies, and transfer learning has contributed to its success in various NLP applications. T5 also has positional encoding to encode the positional information of the input sequence. This positional encoding helps the model understand the order and position of tokens in the sequence, which is crucial for capturing the ordinal nature of the language. Instead of having task-specific architectures, this model considers all NLP tasks as a text-to-text mapping problem. This means that input and output are treated as text strings, allowing T5 to perform various tasks using a unified framework. This approach reduces the complexity of developing and maintaining separate models for each task [11].
- ***BART:*** This is a pre-trained model consisting of Bidirectional and Auto-Regressive transformers. BART is a denoising autoencoder built with a sequence-by-sequence model and can be used for a wide range of end-to-end tasks (prediction, generation, etc.) [12]. Pretraining has two steps: first, the text is corrupted with an arbitrary noise function, and then a sequence-by-sequence model is trained to reconstruct the original text. BART uses a standard transformer-based neural machine translation architecture that, despite its simplicity, can be used as a generalization of BERT (due to bidirectional encoding), GPT (due to left-to-right decoding), and many other pre-trained schemes.

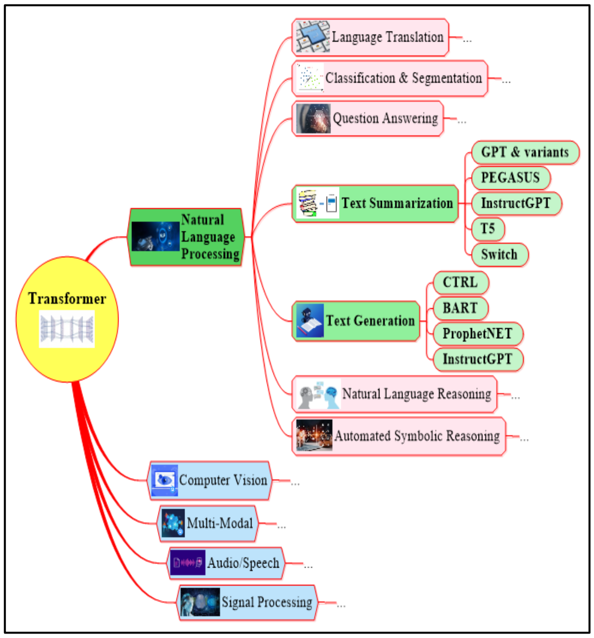

Figure 2. Division of transformer models [9].

2.3. Evaluation measures

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics used to evaluate the quality of summaries and generated text, particularly in natural language processing tasks like automatic summarization and title generation. In line with contemporary research in summarization and text generation, we employ the ROUGE measures for evaluation. In the context of scientific title generation, ROUGE metrics are crucial for evaluating the quality of automatically generated titles reflect the essence of the original articles. Here's why they are important:

- ***Quality Assessment:*** They provide quantitative measures to assess the quality of generated titles against human-written titles.
- ***Comparative Analysis:*** Different models or algorithms can be compared based on their ROUGE scores to determine which one produces better titles.
- ***Model Tuning:*** ROUGE scores can guide the tuning and optimization of title generation models, helping to improve performance iteratively.

While ROUGE is widely used, it has some limitations:

- ***Surface-Level Matching:*** ROUGE primarily measures surface-level overlap and may not capture semantic similarity well.
- ***Sensitivity to Reference Quality:*** The quality and number of reference titles can significantly affect ROUGE scores.
- ***Lack of Context Understanding:*** ROUGE does not account for context or nuances in meaning, which can lead to misleading evaluations.

ROUGE metrics quantify the overlap of units such as n-grams, word sequences, and word pairs between the summaries generated by the machine (to be evaluated) and the ideal human-generated summaries. ROUGE-N specifically measures the n-gram overlap between the generated summary and a predefined set of reference summaries (*RSum*), as defined by (1) [21]:

$$ROURGE-N = \frac{\sum_{s\in\{RSum\}}\sum_{ngram\in S} Count_{match}(ngram)}{\sum_{s\in\{RSum\}}\sum_{ngram\in S} Count(ngram)} \quad (1)$$

In (1), *ngram* represents reference grams, and $Count_{match}$ is the maximum number of overlaps between grams in candidate and reference summaries. The denominator of *ROUGE-N* counts the number of grams that match the candidate and the reference, and the fraction's denominator counts all reference grams.

ROUGE-L uses the longest common subsequence (LCS) and *F-measure* to estimate the similarity between two summaries: the Candidate Summarize *S_can* with length *l_a* and the Reference Summarize *S_ref* with length *l_e* [15]. The calculations related to *ROUGE-L* are given in (2), (3), and (4).

$$recall_{lcs} = \frac{LCS(S_{ref}, S_{can})}{l_e} \quad (2)$$

$$precision_{lcs} = \frac{LCS(S_{ref}, S_{can})}{l_a} \quad (3)$$

$$ROUGE-L = \frac{recall_{lcs} \times precision_{lcs}}{recall_{lcs} + precision_{lcs}} \quad (4)$$

ROUGE-Lsum divides the text into sentences based on newlines (based on the \n character), calculates *LCS* for each pair of sentences, and calculates the average score for all sentences. In the end, we consider ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum measures and the length of the generated title (Gen_Len) as evaluation measures. In the end, we consider ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum measures and the length of the generated title (Gen_Len) as evaluation measures.

The values range of all ROUGE measures are from 0 to 1, where 0 indicates no overlap, and 1 indicates perfect overlap. In practice, scores are often expressed as percentages (e.g., a ROUGE score of 0.4 may be reported as 40).

However, it's important to complement ROUGE measures with other qualitative assessments to gain a comprehensive understanding of title quality. We have utilized expert opinions from each field alongside the ROUGE metrics to evaluate the quality of our generated titles. This way, we aim to address the limitations of the ROUGE metric and achieve a better assessment of the results.

2.4. Literature Review

In the context of title generation, as a branch of summarization, transformer methods is usually used to achieve better results. Ting Zhang et al. [13] presented a method to automatically generate the title of pull requests (PR) related to Github. This method used a dataset containing 43,816 PRs from 495 GitHub repositories. BART, T5, and BERTSumExt models were used to generate automatic PR titles, and the BART method performed better than other methods, with a significant difference. This approach obtained values of 47.22, 25.27, and 43.12 for Rouge-1, Rouge-2, and Rouge-L measures, respectively.

Fengji Zhang et al. [14] presented a method for generating titles for Stack Overflow posts using advanced transformers. They used a big dataset of 890,000 Stack Overflow posts containing questions about coding in eight programming languages. The method used is M3NSCT5, which is based on the T5 method. Finally, Rouge-1, Rouge-2, and Rouge-L measure for eight programming languages: Python: 35.58, 13.28 and 33.03; C#: 31.75, 12.14 and 30.09; Java: 32.94, 11.81 and 30.90; JavaScript: 33.64, 11.71 and 31.65; PHP: 34.53, 11.46 and 31.84; C: 31.22, 10.73 and 29.43; Ruby: 34.77, 13.14 and 32.70; Go: 33.60, 12.23 and 31.74 were obtained respectively.

Also, in another research [15], Fengji Zhang et al. researched generating question titles in Stack Overflow to improve Stack Overflow question titles. This method used a dataset containing 200,000 high-quality Stack Overflow questions. The approach used is CCBERT, which is based on BERT. The CCBERT approach achieved the Rouge-1, Rouge-2, and Rouge-L measures for questions related to the Java programming language, with values of 43.06, 21.15, and 41.76, respectively. For Python programming language questions, the values are 46.69, 22.50 and 44.86, respectively; For the JavaScript programming language questions, get the values of 44.53, 21.05 and 42.80, respectively; and finally, for the PHP programming language questions, get the values of 45.60, 22.35 and 43.87 respectively.

Ting Zhang et al. [16] introduced an approach for automatically generating the title of GitHub Issues, which suggested suitable titles to users. The dataset used in this research is 267,094 GitHub issues, and their approach is iTiger. This approach is based on BART and obtained the values of 40.67, 20.60, and 37.26 for the Rouge-1, Rouge-2, and Rouge-L measures, respectively, for generating issue titles.

Ke Liu et al. [17] presented a transformer-based approach for generating Stack Overflow post titles. The dataset is 284,000 high-quality question posts for four programming languages: Java, C#, Python, and JavaScript. The team's SOTitle approach is based on a well-tuned T5 pre-trained model. Finally, Rouge-1, Rouge-2, and Rouge-L measures for Java programming language are 29.32, 10.98, and 27.28, respectively; For C# programming language, 29.55, 11.96, and 27.61 respectively; 31.84, 11.96, and 29.28 were obtained for Python programming language and 31.15, 11.82 and 28.82 respectively for JavaScript programming language.

Shehab Abdel-Salam and Ahmed Rafea [18] proposed an approach for text summarization called SqueezeBERT. They used CNN.DM dataset and obtained the Rouge-1, Rouge-2, and Rouge-L measures of 43.23, 20.24, and 39.63 for the BERT-base. They obtained values of 42.54, 19.53, and 38.86 for DistilBERT and 42.51, 19.56, and 38.92 for SqueezeBERT. Table 1 shows recent research in title generation and summarization, along with the methods, measures, and datasets used.

3. Proposed method

Our research aims to present a practical method for preparing the title of scientific articles based on the abstract section of the articles. We have used transformer models for our proposed method. For this purpose, after examining different models, in our proposed method, we have used a combination of BART, T5, and finally, PEGASUS models (all based on the basic model) in the form of a particular routine for different steps on the collected data set. In the following, we will introduce the proposed architecture and details of implementing the proposed method.

Table 1. Recent researches in the title generation by transformer methods.

Reference	Year	Method	Dataset	* Measure
[13]	2022	BART	43,816 PR titles on 495 GitHub repositories	Rouge-1 = 47.22, Rouge-2 = 25.27, Rouge-L = 43.12
[14]	2023	M3NSCT5	890,000 question posts on Stack Overflow	Rouge-1 = 35.58, Rouge-2 = 13.28, Rouge-L = 33.05
[15]	2022	CCBert	200,000 questions on Stack Overflow	Rouge-1 = 47.03, Rouge-2 = 23.50, Rouge-L = 45.15
[16]	2022	iTAPE, iTIGER	267,094 GitHub Issues	Rouge-1 = 40.67, Rouge-2 = 20.6, Rouge-L = 37.26
[17]	2022	SOTitle	284,000 posts of programming languages on Stack Overflow	Rouge-1 = 31.82, Rouge-2 = 11.98, Rouge-L = 29.28
[18]	2022	BERT-base	CNN/DM Dataset	Rouge-1 = 43.23, Rouge-2 = 20.24, Rouge-L = 39.63

* These Results are the maximum results for these experiences.

3.1. Data set

This section discusses how to prepare the data set and related challenges. The raw and unprocessed dataset was extracted from the ScienceDirect.com website, which contains an extensive collection of scientific and research articles in various fields of science (technical and engineering, medicine, humanities, natural sciences, economic sciences, etc.). The articles used have been selected from various scientific fields, and they include articles whose publication date is between 2020 and 2023. From each article, only their title and abstract are considered. The possibility of a direct search of the ScienceDirect.com website has been used to extract the title and abstract of the articles.

After the data extraction phase, the challenge in the dataset preparation phase is the possibility of miss values, and especially the abstract section not being the same in different articles. (According to the types of the articles, the abstract section may be one section or contain several sections such as Introduction, Methods, Result, and Conclusion sections). We only used articles whose abstract section was one part. Finally, we have collected all the extracted data in a dataset with two attributes, Abstract and Title, for each item. To achieve the appropriate volume, 50,000 articles were collected after preprocessing. The prepared dataset is available in GitHub[1].

3.2. Structure of the proposed method

Fig. 3 shows the architecture of our proposed method. In general, this research used an approach that links the results of the PEGASUS, T5, and BART models separately with TextRank. Then, their output is sent separately to other PEGASUS, T5, and BART models. The reason for this action is based on our finding that general models like BERT, T5, and PEGASUS are optimized for constructing complete sentences, but their ability to generate scientific titles is more limited. By training the models in two stages on the prepared data, our goal is to refine the models' focus on the structure of scientific titles, resulting in a significant improvement in the quality of generated content. Based on the results of various efforts to generate article titles, it was found that in the first stage, the generated title consists solely of important keywords from the original text. In the second stage, by presenting the outcome of the first stage along with the original text, the generated title is significantly improved. In the following, we will describe the implementation steps of the proposed method in detail.

3.2.1. Preprocessing

After preparing the data set, preprocessing is done on the data set. Figure 4 shows the preprocessing steps that include the following tasks: In the first step, all letters are converted to lowercase. Then, to ensure that the title or abstract field was not mistakenly null during the data collection stage, the records with a null value in the title or the abstract are deleted. In the third step, the duplicate records are removed. Next, the HTML tags are removed. Finally, signs and punctuation marks are removed (except point signs).

As the title of the article may contain numbers and abbreviations, the decision was made to refrain from removing numbers, abbreviations, and stop words. After these steps, final dataset has 50,000 records. The dataset is divided into 60% (i.e. 30,000 records) training data, 20% (i.e. 10,000 records) evaluation data, and 20% (i.e. 10,000 records) experimental data.

[1] https://github.com/mohammadpur/PTRP/blob/main/PTRP.csv

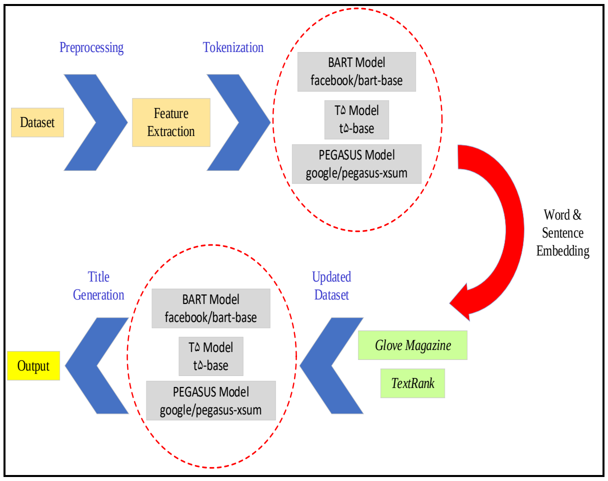

Figure 3. Architecture of the proposed model.

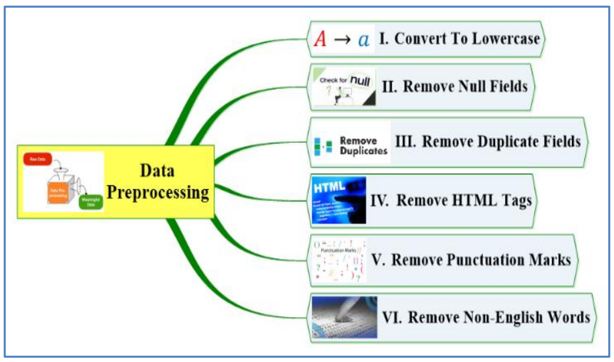

Figure 4. Preprocessing stages.

3.2.2. Tokenization

In this step, we identify and separate a list of tokens and send it to the tokenizer. Given that summarization and title generation rely on sequence-by-sequence methods, the model utilized for this purpose must adhere to a sequence-by-sequence architecture. Additionally, AutoTokenizer from the same model is employed for tokenization.

As a result of varying token lengths, padding was implemented to standardize the token size. The maximum token length served as the reference, ensuring all tokens matched this length. Subsequently, for batch processing, titles and abstracts were categorized as Key-Value pairs. This format was utilized for training, evaluation, and testing data processing into DataFrame form.

3.2.3. Implementation of basic models

As previously stated, this study employs BART, T5, and PEGASUS transformer models to generate titles for scientific articles. Upon loading these models, computations pertaining to training and evaluation are carried out until the entire training process is finalized. Subsequently, the trained model is saved, and the experimental data is assessed using the saved model.

3.2.4. Embedding and TextRank

During this stage, an initial prediction is formulated utilizing the preserved model while taking into account a masked attention layer. Subsequently, the word embedding process is initiated. At this juncture, the Glove word embedding is imported, featuring a compilation of word vectors trained on six billion words sourced from the Wikipedia and Web Text repositories. Each entry comprises a word paired with the corresponding values of its hundred-dimensional vector. Following the word embedding phase, the TextRank technique is employed to prioritize the sentences within the abstracts of the articles [19].

The TextRank algorithm is characterized as an unsupervised extractive text summarization technique. Within the TextRank framework, the entirety of the document's text is amalgamated, subsequently segmented into individual sentences. Following this segmentation, the word embedding process is executed for each sentence, generating the corresponding word vectors. The similarity between these word embeddings is computed and recorded in a similarity matrix. This matrix is transformed into a graph structure, with sentences acting as nodes and similarity scores serving as edges. Through this iterative process, sentences with the highest ranking are selected for inclusion in the final summary [20].

3.2.5. Implementation of the second model

In the subsequent stage, a novel column titled "second_abstract" is appended to the dataset, containing the following information: initially, the primary title derived in the preceding phase is placed, followed by arranging sentences from the articles' abstracts based on their TextRank score. Subsequently, the "second_abstract" column is integrated as an additional abstract column in the train, validation, and test datasets. Post the tokenization procedure, the refined dataset is inputted into the model. The second model mirrors the first model, operating with identical parameters and configurations. Upon model execution, the outcomes are evaluated.

4. Results and evaluation

As previously stated, we generated titles for scientific articles through the creation of a unique and tailored dataset, employing meticulous preprocessing techniques, and ultimately leveraging transformer models (BART, T5, and PEGASUS) in conjunction with the TextRank method. The subsequent section delineates and assesses the outcomes derived from the application of the aforementioned methodologies.

Table 2. Specifications and parameters of BART, T5, and PEGASUS models.

BART	T5	Pegasus
vocab_size =50265	vocab_size = 32128	vocab_size = 50265
max_position_embeddings = 1024	d_model = 512	max_position_embeddings = 1024
encoder_layers = 12	d_kv = 64	encoder_layers = 12
encoder_ffn_dim = 4096	d_ff = 2048	encoder_ffn_dim = 4096
encoder_attention_heads = 16	num_layers = 6	encoder_attention_heads = 16
decoder_layers = 12	num_decoder_layers = None	decoder_layers = 12
decoder_ffn_dim = 4096	num_heads = 8	decoder_ffn_dim = 4096
decoder_attention_heads = 16	relative_attention_buckets = 32	decoder_attention_heads = 16
encoder_layerdrop = 0	relative_attention_distance = 128	encoder_layerdrop = 0
decoder_layerdrop = 0	dropout_rate = 0.1	decoder_layerdrop = 0
activation_function = 'gelu'	layer_norm_epsilon = 1e-06	use_cache = True
d_model = 1024	initializer_factor = 1.0	is_encoder_decoder = True
dropout = 0.1	feed_forward_proj = 'relu'	activation_function = 'gelu'
attention_dropout = 0	is_encoder_decoder = True	d_model = 1024
activation_dropout = 0	use_cache = True	dropout = 0.1
init_std = 0.02	pad_token_id = 0	attention_dropout = 0
classifier_dropout = 0	eos_token_id = 1	activation_dropout = 0
scale_embedding = False	classifier_dropout = 0	init_std = 0.02
pad_token_id = 1		decoder_start_token_id = 0
eos_token_id = 2		scale_embedding = False
is_encoder_decoder = True		pad_token_id = 0
decoder_start_token_id = 2		eos_token_id = 1
forced_eos_token_id = 2		forced_eos_token_id = 1

4.1. Model settings

Given the utilization of BART and T5 transformer models in recent research on summarization and text generation [13], [14], [16], [17], we have similarly employed the basic versions of these models, aligning our parameters with the optimal settings identified in the literature. Furthermore, we have harmonized the parameter configurations of the PEGASUS model to maintain consistency. Table 2 encapsulates the specifications and parameters of the BART, T5, and PEGASUS models, focusing on their fundamental characteristics. These specifications closely resemble those of the BART and PEGASUS models.

In the proposed model, following thorough experimentation with various settings, we have identified the most appropriate values for batch size, number of iterations, number of steps, and other variables, as detailed in Table 3. Given the utilization of three distinct transformer models (BART, T5, and Pegasus), these values have been uniformly applied with identical conditions across all three models.

4.2. Checking the results

The final results derived from the implementation of the proposed method, as per the specified evaluation criteria, are presented in Tables 4, 5, and 6. Table 4 displays the outcomes of the BART_TextRank_BART (BTRB) model, Table 5 shows the results of the T5_TextRank_T5 (T5TRT5) model, while Table 6 exhibits the findings of the Pegasus_TextRank_Pegasus (PTRP) model.

Table 3. Settings of the proposed model.

Setting Title	Value
seed	42
data_seed	42
num_train_epochs	4
do_train	True
do_eval	True
per_device_train_batch_size	2
per_device_eval_batch_sizee	2
warmup_steps	200
weight_decay	0.05
label_smoothing_factor	0.05
predict_with_generate	True
logging_dir	Logs
logging_steps	15000
evaluation_strategy	Steps
save_total_limit	4
save_strategy	Steps
save_steps	15000
load_best_model_at_end	True

Table 4. The results of the BTRB model

Step	Rouge-1	Rouge-2	Rouge-L	Rouge-Lsum	Gen Len
15,000	50.81	29.88	44.23	44.29	18.47
30,000	50.74	29.82	44.18	44.23	18.42
45,000	50.76	29.83	44.18	44.23	18.47
60,000	50.78	29.83	44.19	44.24	18.48

Table 5. The results of running the T5TRT5 model

Step	Rouge-1	Rouge-2	Rouge-L	Rouge-Lsum	Gen Len

15,000	49.43	28.06	42.43	42.51	16.65
30,000	49.66	28.26	42.68	42.76	16.71
45,000	49.31	27.87	42.35	42.44	16.71
60,000	49.44	27.81	42.31	42.39	16.95

Table 6. The results of PTRP model implementation.

Step	Rouge-1	Rouge-2	Rouge-L	Rouge-Lsum	Gen Len
15,000	52.13	30.58	45.08	45.15	15.92
30,000	52.35	30.80	45.31	45.39	16.04
45,000	52.36	30.81	45.33	45.40	16.05
60,000	52.30	30.72	45.24	45.31	16.16

Furthermore, Figure 5 illustrates the training and evaluation errors chart for the BTRB model, Figure 6 depicts the corresponding chart for the T5TRT5 model, and Figure 7 showcases the training and evaluation errors chart for the PTRP model.

4.4. Comparative Analysis of BTRB, T5TRT5, and PTRP Methods Utilizing Experimental Data

Upon executing the BTRB, T5TRT5, and PTRP methodologies on the test dataset, the metrics for Rouge-1, Rouge-2, Rouge-L, Rouge-Lsum, and Gen-Len are detailed in Table 7. The findings demonstrate that the proposed approach outperformed prior studies by incorporating all three models (BART, T5, and PEGASUS), as illustrated in Table 1. Furthermore, according to the outcomes presented in Table 7, the PTRP model exhibited superior performance compared to the alternative methods.

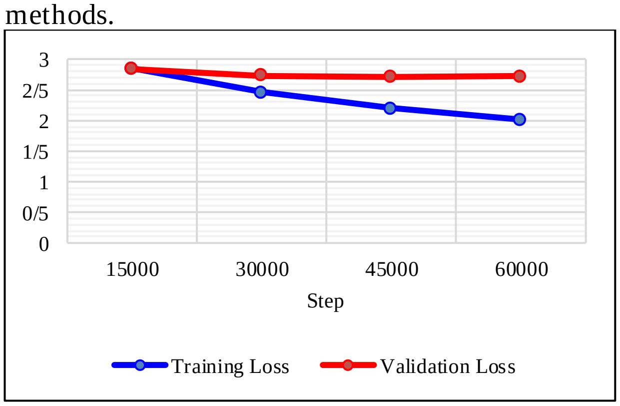

Figure 5. Training and evaluation errors chart of the BTRB model on the training data set.

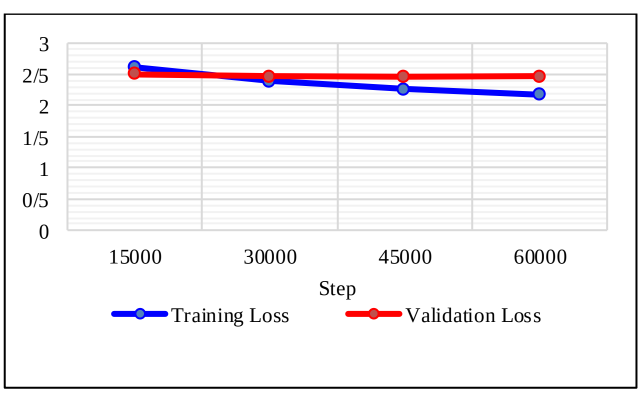

Figure 6. Training and evaluation errors chart of the T5TRT5 model on the training data set.

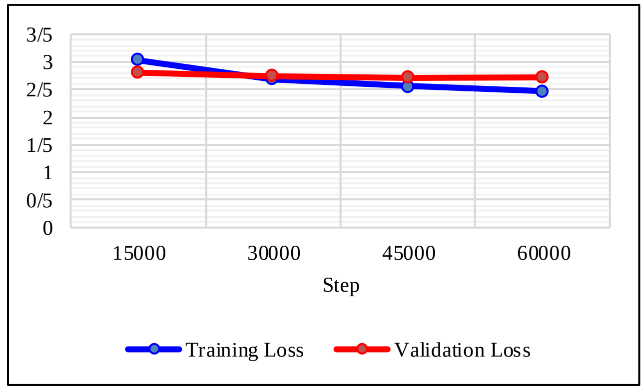

Figure 7. Training and evaluation errors chart of the PTRP model on the training data set.

Table 7. The results of running PTRP, T5TRT5, and BTRB models on the test data set.

Model	Rouge-1	Rouge-2	Rouge-L	Rouge-LSum	Gen_Len
BTRB	50.92	30.06	44.52	44.55	18.36
T5TRT5	51.07	30.09	44.48	44.52	17.20
PTRP	**54.87**	**33.61**	**47.89**	**47.94**	**16.96**

Table 8. A Sample of output from the Execution of BTRB, T5TRT5, and PTRP.

Abstract of the article:
Podcasting is used in higher education to share various digital resources with students. This review aims to synthesize evidence on podcasting in nursing and midwifery education. PubMed, Medline, Cinahl, Scopus, and Eric databases were searched using key terms. Two hundred forty-two articles were found and screened. Data extraction, quality assessment, and data analysis, underpinned by a social media learning model, were conducted on relevant studies. Twenty-six studies were included in the review. Three themes emerged: 1) learning and other outcomes, 2) antecedents to learning, and 3) learning process. Students acquired new knowledge and skills by using podcasts, and it also improved clinical confidence. The organization of podcasting, digital literacy, e-professionalism, learners' motivation, and flexible access to technology impacted the delivery of this educational intervention. Mechanisms that affected the learning process were the speed of exchange, the type of social media user, the timeframe, the quality of information, the functionality of podcasts, and other learning activities. This review synthesized evidence of podcasting in nursing and midwifery education. The technology was seen as a positive learning tool, but more robust research is needed to examine its efficacy in improving learning outcomes.
The title of the article:
Podcasting in nursing and midwifery education: an integrative review
Title generated using the BTRB model:
Digital entrepreneurship in nursing and midwifery education: a systematic review
Title produced using T5TRT5 model:
A review of podcasting in nursing and midwifery education

Title generated using PTRP model:
Podcasting in nursing and midwifery education: a systematic review

4.5. Analyzing the results of the models

We utilized a large number of outputs from our proposed models for generating article titles and evaluated the obtained outputs from the perspective of an expert person in each field. The evaluation results indicate that the proposed methods, especially the PTRP method, have successfully generated acceptable titles for scientific articles. For example, a sample of the outputs is provided in Table 8.

Table 8 presents example of article titles generated from the abstract utilizing the BTRB, T5TRT5, and PTRP models. As it is seen, the generated titles demonstrate exceptional quality.

Nevertheless, to enhance and refine the outputs further, an expert person evaluation is imperative. This evaluation entails experts and researchers from respective scientific domains assessing the generated titles by appraising the articles in their specific fields. And, if the quality of the output is low in a certain field, by strengthening the dataset related to that field, it is possible to produce more suitable outputs.

5. Conclusion

The title of a text should encapsulate the essence of the content concisely and accurately, using minimal terminology. Our study introduces innovative models derived from foundational text-processing frameworks. The findings demonstrate the potential of deep learning techniques, particularly transformer models, in automating summarization and title generation, especially for scientific articles. Models like BERT, T5, and PEGASUS are optimized for constructing complete sentences, but their ability to generate scientific titles is more limited. By training the models in two stages on the prepared data, we aim to direct the models' focus towards the structure of scientific titles, resulting in a significant improvement in output. The empirical evidence highlights the superior performance of the PTRP model, combining PEGASUS and TextRank. This research emphasizes the critical impact of a comprehensive and well-structured dataset on achieving optimal outcomes. While quantity plays a role, the quality of the dataset is paramount. Furthermore, we have created a diverse dataset of article titles and abstracts across various scientific disciplines, providing a valuable resource for future investigations in this field. The PTRP method has shown its effectiveness in suggesting suitable titles based on article abstracts. Comparing these results with previous methods reveals approximately 4% improvement based on various ROUGE metrics. Moreover, the positive evaluation by expert persons confirms the high quality of the results and validates the PTRP method as a highly efficient approach.

References

[1] N. Nazari, and M. A. Mahdavi, "A survey on automatic text summarization," *Journal of AI and Data Mining*, vol. 7, no. 1, pp. 121-135, 2019.

[2] D. Radev, E. Hovy, and K. McKeown, “Introduction to the special issue on summarization," *Computational linguistics*, vol. 28, no. 4, pp. 399-408, 2002.

[3] M. Zhang, G. Zhou, W. Yu, N. Huang, and W. Liu A, "Comprehensive survey of abstractive text summarization based on deep learning," *Computational intelligence and neuroscience*, vol. 2022, no. 1, pp. 1-21, 2022.

[4] S. Mehrabi, S. A. Mirroshandel, and H. Ahmadifar, “DeepSumm: A Novel Deep Learning-Based Multi-Lingual Multi-Documents Summarization System," *Information Systems & Telecommunication (JIST)*, vol. 7, pp. 204-214, 2019.

[5] K. Kaku, M. Kikuchi, T. Ozono, and T. Shintani, “Development of an extractive title generation system using titles of papers of top conferences for intermediate English students," in *Proceedings of the 10th International Congress on Advanced Applied Informatics*, 2021, pp. 59-64.

[6] W. Li, X. Xiao, Y. Lyu, and Y. Wang, “Improving neural abstractive document summarization with explicit information selection modeling," in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 1787-1796.

[7] M. Molaei, D. Mohamadpur, "Distributed Online Pre-Processing Framework for Big Data Sentiment Analytics," *Journal of AI and Data Mining*, vol. 10, pp. 197-205, April 2022.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need," in *Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS)*, 2017, pp. 10-21.

[9] S. Islam, H. Elmekki, A. Elsebai, J. Bentahar, N. Drawel, and W. Pedrycz, “A comprehensive survey on applications of transformers for deep learning tasks," *Expert Systems with Applications*, vol. 241, pp. 122666, 2024.

[10] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, “Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," in *Proceedings of the International conference on machine learning*, 2020, pp. 11328-11339.

[11] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, pp. 1-67, 2020.

[12] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint*, arXiv:1910.13461, 2019.

[13] T. Zhang, I. C. Irsan, F. Thung, D. Han, D. Lo, and L. Jiang, "Automatic pull request title generation," in *Proceedings of the IEEE International Conference on Software Maintenance and Evolution*, 2022, pp. 71-81.

[14] F. Zhang, J. Liu, Y. Wan, X. Yu, X. Liu, and J. Keung, "Diverse title generation for Stack Overflow posts with multiple-sampling-enhanced transformer," *Journal of Systems and Software*, vol. 200, pp. 111672, 2023.

[15] F. Zhang, X. Yu, J. Keung, F. Li, Z. Xie, Z. Yang, C. Ma, and Z. Zhang, "Improving Stack Overflow question title generation with copying enhanced CodeBERT model and bi-modal information," *Information and Software Technology*, vol. 148, pp. 106922, 2022.

[16] T. Zhang, I. C. Irsan, F. Thung, D. Han, D. Lo, and L. Jiang, "iTiger: an automatic issue title generation tool," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2022, pp. 1637-1641.

[17] K. Liu, G. Yang, X. Chen, and C. Yu, "Sotitle: A transformer-based post title generation approach for stack overflow," in *Proceedings of the IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2022, pp. 577-588.

[18] S. Abdel-Salam, and A. Rafea, "Performance study on extractive text summarization using BERT models," *Information*, vol. 13, pp. 67, 2022.

[19] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing*, 2014, pp. 1532-1543.

[20] S. Bhargav, A. Choudhury, S. Kaushik, R. Shukla, and V. Dutt, "A comparison study of abstractive and extractive methods for text summarization," in *Proceedings of the International Conference on Paradigms of Communication, Computing and Data Sciences*, 2022, pp. 601-610.

[21] C. Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proceedings of the Text summarization branches out*, 2004, pp. 74-81.

تولید عنوان بر اساس مدل‌های ترنسفورمر

داود محمدپورزنجانی* و مهدی نظری

گروه آموزشی کامپیوتر، دانشگاه زنجان، زنجان، ایران.

ارسال ۲۰۲۴/۰۶/۱۸؛ بازنگری ۲۰۲۴/۰۹/۰۵؛ پذیرش ۲۰۲۴/۱۰/۱۸

چکیده:

خلاصه‌سازی متون به دلیل رشد سریع مطالب و محتواهای مختلف به یکی از موضوعات مورد علاقه محققین تبدیل شده است. یکی از زیرمجموعه‌های کلیدی خلاصه‌سازی متن، ایجاد یک عنوان مناسب و معنادار برای متن می‌باشد که از اهمیت بسیاری برخوردار است. از آن جهت که یک عنوان مناسب می‌تواند منعکس کننده محتوا، اهداف، روش‌ها و یافته‌های متن باشد، تولید یک عنوان مناسب مستلزم درک کامل متن است. روش‌های مختلفی در خلاصه‌سازی متن برای تولید خودکار عناوین پیشنهاد شده‌اند. ما در این مقاله یک روش تولید عنوان برای مقالات علمی با استفاده از روش‌های مبتنی بر ترنسفورمرها ارائه نموده‌ایم که برای تولید عنوان مقاله از چکیده مقاله استفاده می‌نماید. مدل های مبتنی بر ترنسفورمرها که عمدتا با عنوان مدل‌های از قبل آموزش دیده شده مانند BERT، T5 و PEGASUS شناخته می‌شوند برای ساخت جملات کامل بهینه شده‌اند، اما توانایی آنها برای تولید عناوین علمی محدود است. ما سعی کرده‌ایم این محدودیت را با ارائه روش پیشنهادی که مدل‌های مختلف را با استفاده از یک مجموعه داده مناسب آموزش می‌دهد، بهبود ببخشیم. این مجموعه داده را براساس چکیده‌ها و عناوین مقالات منتشر شده در وب سایت ScienceDirect.com ایجاد نموده‌ایم. پس از انجام پردازش‌ها، یک مجموعه داده مناسب متشکل از ۵۰۰۰۰ مقاله ایجاد کرده‌ایم. نتایج حاصل از ارزیابی روش پیشنهادی حاکی از بهبود ۴ درصدی بر اساس معیارهای ROUGE در تولید عناوین علمی نسبت به روش‌های مشابه است. علاوه بر این، بررسی عناوین تولید شده توسط متخصصان هر رشته علمی نشان می‌دهد که عناوین مورد قبول این متخصصان می‌باشد.

کلمات کلیدی: خلاصه‌سازی متن، تولید خودکار عنوان، ترنسفورمر، مدل از قبل آموزش دیده.