Iraqi Journal for Computers and Informatics
Available online: https://ijci.uoitc.edu.iq/index.php/ijci
Volume 52, issue 1. 2026

Print ISSN: 2313-190X, Online ISSN: 2520-4912

Keywords-to-Title model for Automated Academic Title Generation

Ali Ahmed Ali [a], Mohammed Ali Mohammed [b]

[a] *Informatics Institute for Postgraduate Studies, University of Information Technology and Communications (UOITC), Baghdad, Iraq.*
[b] *College of Business Informatics, University of Information Technology and Communications (UOITC), Baghdad, Iraq.*

ARTICLE INFO

Keywords:

Generate Title, NIPS dataset, Training and Testing, Keywords Extraction.

ABSTRACT

In an article, the title should cover the whole content with a few important words. Several automated title-generating tools are available in the Internet. This paper is aiming to design and implementation a new model to generate title using a list of keywords. The model using a new dataset that is generated from NIPS dataset with Configure training arguments. The proposed system preprocesses keyword data, trains on a curated dataset, and produces coherent, contextually relevant titles through controlled text generation. The proposed model shows a strong generative capability by accurately producing research titles from list of important keywords. Its efficient fine-tuning strategy enables high performance with minimal training resources. The experimental results show that the proposed fine-tuned T5 title generation model can produce titles that are very close to the original scientific titles. For the paper "Learning to Play the Game of Chess", the generated title matched the original exactly. As a result, all evaluation metrics reached 1.0, including cosine similarity, ROUGE-L, BERT_F1, and SciBERT_F1, indicating complete lexical and semantic agreement between the generated and reference titles.

1. INTRODUCTION

Automated title generation is a newly emerging field, falling under the umbrella of natural language processing (NLP). The development of these techniques aims to provide a comprehensive summary of the research gist and include it as the title of the research paper, thus presenting the content of the research through the automatically generated sentence. It is worth noting how this method can help researchers in various fields improve their research capabilities and discover academic works [1], [2]. To ensure the relevance and clarity between the generated title and the content of the research paper, multiple linguistic and statistical techniques are used [3], [4].

The title is considered a front and summary of the content of a research paper, and it is used to infer the content of the research or article [5]. Many authors seek to index previously written good titles to develop appropriate techniques for automated title writing [6]. The extractive summarization technique has been used in the news field [6][7][8]. Based on the current reality, there is a lot of information that can be extracted from the article's objectives and the meaning that the author wants to convey, but extracting the titles was based on extracting the related words and forming the title without considering the structural structure of the article [9].

Training a model involves fitting a parametric computational architecture (e.g., a deep neural network) on labeled data to learn a generalizable mapping from inputs to outputs. In this process, the model learns from the data so that it can handle tasks such

E-mail address:
ms202410015@iips.edu.iq [a]
mohammed.ali@uoitc.edu.iq [b]

Corresponding* : *Ali Ahmed Ali*

Received 26 December2025, Accepted 17 February 2026

DOI: 10.25195/ijci.v52i1.714 .

as text generation and image recognition. This makes deep learning useful in many areas [10]. However, training these models is not easy. They usually need a large amount of high-quality labeled data, strong computational resources, and a long training time. They are also hard to interpret, which can make errors harder to understand [11]. Therefore, effective model training depends on good data, enough resources, and suitable evaluation methods to reduce overfitting and bias.

Given the above context, the following subsections present the problem statement, research gap, and main contributions. With this above setting, the problem statement, research gap, and key contributions are as shown in the following subsections.

Problem Statement: Although academic paper titles play a central role in increasing their visibility and enabling fast understanding, the process of writing concise and truthful titles is tedious and gives way to inconsistencies. Solutions to this issue can be provided by the existing automated methods, which give titles that are clumsy, verbose or poorly aligned to the research interest of the intended study where the model input does not directly specify which important concepts to include in the title.

Research Gap: In spite of some studies using a strategy of summarizing salient sentences in abstracts, before generating titles, many studies are still using full-text abstract or generic text as the input to a model to generate the title. Moreover, end-to-end pipelines that integrate summarization-driven keyword construction, domain-weighted keyword selection, controlled keyword-to-title generation, and semantic evaluation are not consistently presented in a reproducible manner, which can reduce title faithfulness and consistency.

Contributions: This paper contributes the following:

1) A dataset construction pipeline that derives six weighted keywords per paper from NIPS/NeurIPS papers and pairs them with the original titles for supervised training.

2) A keyword-to-title generation model based on fine-tuning T5-base to map compact keyword inputs into fluent academic titles.

3) A comparative evaluation against KeyToText and GPT-2 using both lexical and semantic metrics (Cosine similarity, ROUGE-L, BERTScore-F1 (BERT/SciBERT)).

The paper is organized as follows: Section 2 describes the dataset construction. Section 3 presents the methodology of the proposed model. Section 4 reports the results and discussion. Finally, Section 5 concludes the paper.

2. CREATE A DATASET

In this section, a new dataset will be created according to our model requirements. Neural Information Processing Systems (NIPS) [12] is one of the top machine learning conferences in the world. It covers topics ranging from deep learning and computer vision to cognitive science and reinforcement learning. This dataset includes the title, authors, abstracts, and extracted text for all NIPS papers to date (ranging from the first 1987 conference to the current 2016 conference). Extracted the paper text from the raw PDF files and are releasing that both in CSV files and as a SQLite database [12][13].

The structure of a new dataset contains a title and keywords which is used to build our model in next section. To build it, for each paper in the NIPS dataset, read the abstract, apply preprocessing steps (such as case normalization [15], noise removal [16], tokenization [9][15], stop-word removal [17], and lemmatization [14] ensures clean, analyzable input for NLP tasks.,), then apply the luhn summarization [18] algorithm, finally, apply the modify Yake algorithm to extract a six weighted words. in last, a new dataset saved as original title (NIPS title) and six keywords extracted. Table 1 summarizes the dataset construction settings, and Table 2 shows the sample of a new dataset.

Table 1: Dataset description and construction settings

Item	Description
Source dataset	NIPS/NeurIPS papers dataset (Kaggle) [12]
Raw file (used fields)	papers - Copy.csv (title, paper_text)
Abstract extraction	Abstract text is extracted from paper_text using heuristic markers (abstract → introduction); records without a detected abstract are skipped
Preprocessing	Lowercasing, regex-based noise cleaning, tokenization, stop-word removal (NLTK + custom list), short-token filtering, and WordNet lemmatization
Summarization	Luhn summarization (Sumy); top 3 sentences retained

Keyword construction (Modified YAKE)	YAKE produces ranked candidates; domain weighting is applied using CS_Words.txt with factor cs_boost_factor=0.3; the final top-6 weighted keywords are selected as model input
Output training dataset	Pairs: (original title, six keywords); saved as CS_Dataset.csv

Table 2: Sample of a new dataset

Title	Keywords
Self-Organization of Associative Database and Its Applications	['efficient', 'database', 'method', 'self-organizing', 'associative', 'application']
Using a neural net to instantiate a deformable model	['model', 'recognition', 'imaging', 'method', 'efficient', 'algorithm']
A Mean Field Theory of Layer IV of Visual Cortex and Its Application to Artificial Neural Networks	['cell', 'network', 'layer', 'signal', 'single', 'cortical']
An experimental comparison of recurrent neural networks	['recurrent', 'neural', 'network', 'architecture', 'identification', 'proposed']
Bayesian Query Construction for Neural Network Models	['data', 'model', 'informative', 'sequential', 'bayesian', 'selection']

3. METHODOLOGY OF OUR MODEL

T5 is a transformer-based sequence-to-sequence model that formulates NLP tasks in a unified text-to-text framework [19]. A new model was developed by fine-tuning the T5-base architecture to generate research titles from structured keyword inputs. The experimental environment was set up by detecting available computational resources and loading the pre-trained model and tokenizer. The dataset was loaded in local CSV files generated in the foregoing section. Its integrity was tested by ensuring that the necessary fields were present and then it was preprocessed by converting keywords into a list representation and building input-output pairs to be used in the further training of the model.

The T5-base model uses an encoder-decoder Transformer model. The keyword prompt is fed to the encoder and the decoder produces title based on what is represented. At the fine-tuning stage, the model is trained using the (keywords title) pairs, and this allows the titles obtained to resemble the reference titles that are in the training set.

To carry out evaluation, the created dataset was split into 90 per cent training data and 10 per cent validation data. The HuggingFace Trainer framework was used to train the model with explicitly defined hyperparameters with a learning rate, a batch size, the number of epochs, the number of gradient accumulation steps, and the evaluation strategy. The framework coordinated training, validation, checkpointing, and backpropagation, hence, leading to the generation of a strong fine-tuned model and tokenizer to be deployed further. Table 3 provides the exact hyperparameters used in the experiments and the decoding configurations are those used in title generation.

Table 3: Fine-tuning and decoding configuration (implementation-based)

Parameter	Value
Model checkpoint	t5-base
Train/validation split	90% / 10% (random_state=42)
Epochs	200
Learning rate	5e-4
Batch size (train/eval)	8 / 8
Gradient accumulation steps	2
Warmup steps	500
Weight decay	0.01
LR scheduler	linear
Best model selection	load_best_model_at_end=True, metric_for_best_model=eval_loss

Decoding (inference)	num_beams=4, do_sample=True, temperature=0.7, top_p=0.9, max_length=64, early_stopping=True
Tokenization (max length)	input=128, target=64
Random seed	42

For evaluation, a title-generation function was implemented to produce titles from unseen keyword sets. During inference, keywords were formatted, tokenized, and input into the model, which generated titles using beam search with temperature-controlled sampling and top-p filtering. Empirical testing showed that the model produced coherent, contextually appropriate, and semantically meaningful titles, demonstrating the effectiveness of the fine-tuning approach for automated academic title generation. Algorithm 1 show the steps of generating our model. And figure 1 shows the flowchart of generate our model.

Algorithm 1: Generate our model

Input: Dataset with fields (title, keywords)

Output: Model

Step 1: Environment Initialization
- Detect GPU availability and set PyTorch device.
- Load the T5-base tokenizer and T5-base model from the HuggingFace Transformers library.

Step 2: Dataset Loading
- Load a new dataset file
- Validate required columns: title, keywords.

Step 3: Data Preprocessing

Step 4: Train/Validation Split, divide the processed data into (90% training set, 10% validation set)

Step 5: Tokenization, Tokenize input and target texts using T5-base tokenizer. Add labels field using the tokenized targets.

Step 6: Convert training/validation Pandas dataframes into Dataset objects.

Step 7: Use DataCollatorForSeq2Seq to dynamically pad sequences and prepare model inputs.

Step 8: Set Training Configuration

Step 9: Model Training
- Instantiate a Trainer object.
- Train the model using the tokenized dataset.
- Save the final fine-tuned model and tokenizer.

Step 10: Model Inference (Title Generation): Load the fine-tuned model.

Step 11: Testing
- Evaluate the model using a sample list of keywords.
- Print the generated title.

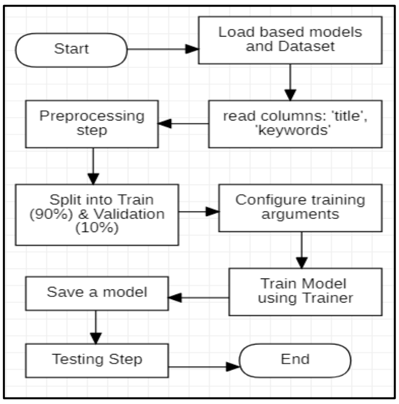

Fig. 1. *Flowchart to generate our model*

4. RESULT AND DISCUSSION

To evaluate the performance of the proposed title generation approach, three models are considered: our fine-tuned T5- model, the KeyToText model, and the GPT-2 model. For all experiments, the abstracts in the dataset are first pre-processed by applying text cleaning, stop-word removal, and lemmatization. Then, the Luhn summarization algorithm is applied to obtain a condensed version of each abstract. After that, the Modified YAKE algorithm is used on the summarized text to extract the most informative keywords, which are then fed into the three models to generate the final titles. Table 4 shows the generated title using our model comparing with original title. Table 5 shows the generated title using KeyToText comparing with original title. Table 6 shows the generated title using gpt-2 comparing with original title. Evaluation was performed using cosine similarity as a statistical measure to quantify the similarity between generated and reference titles based on their vector representations [20][21], along with ROUGE-L [22], BERTScore (BERT_F1) [23], SciBERT_F1 [24], and supporting evidence from recent title-generation studies [25].

Table 4: Result of generated title using our model

No	Keywords_6_Used	Original Title	Generated Title	Evaluation Metrics
1	['program', 'games', 'learns', 'chess', 'learning', 'play']	Learning to Play the Game of Chess	Learning to Play the Game of Chess	Cosine= 1.0 ROUGE-L=1.0 BERT_F1=1.0 SciBERT_F1=1.0
2	['security', 'privacy', 'computing', 'cloud', 'data', 'protection']	Data security and privacy in cloud computing	The Privacy-preserving cloud computing	Cosine= 0.6328 ROUGE-L=0.3077 BERT_F1=0.9217 SciBERT_F1=0.8083
3	['processor', 'real-time', 'wavelet', 'decomposition', 'reconstruction', 'continuous']	Analog VLSI Processor Implementing the Continuous Wavelet Transform	Analog VLSI Processor Implementing the Continuous Wavelet Transform	Cosine= 1.0 ROUGE-L=1.0 BERT_F1=1.0 SciBERT_F1=1.0
4	['system', 'nonlinear', 'parametric', 'uncertainty', 'parameter', 'neural']	Neural Control for Nonlinear Dynamic Systems	Nonlinear Parameter Estimation in Nonlinear Systems	Cosine=0.2884 ROUGE-L=0.3333 BERT_F1=0.9143 SciBERT_F1=0.8297

Table 5: Result of generated title using KeyToText model

No	Keywords_6_Used	Original Title	Generated Title	Evaluation Metrics
1	['program', 'games', 'learns', 'chess', 'learning', 'play']	Learning to Play the Game of Chess	The program "games| learning|"" is "play.	Cosine = 0.3041 ROUGE-L= 0.3077 BERT_F1= 0.8270 SciBERT_F1 = 0.5830
2	['security', 'privacy', 'computing', 'cloud', 'data', 'protection']	Data security and privacy in cloud computing	The security of a cloud with a data protection of 512 m.	Cosine = 0.0585 ROUGE-L = 0.1176 BERT_F1 = 0.8404 SciBERT_F1 = 0.6185
3	['processor', 'real-time', 'wavelet', 'decomposition', 'reconstruction', 'continuous']	Analog VLSI Processor Implementing the Continuous Wavelet Transform	The real-time "wavelet" "reintegration" is "continu.".	Cosine = 0.1559 ROUGE-L = 0.2667 BERT_F1 = 0.8506 SciBERT_F1 = 0.5311
4	['system', 'nonlinear', 'parametric', 'uncertainty', 'parameter', 'neural']	Neural Control for Nonlinear Dynamic Systems	The system "system| nonlinear| parametric|||incided" is logical.	Cosine = 0.0700 ROUGE-L = 0.1429 BERT_F1 = 0.8427 SciBERT_F1 = 0.5248

Table 6: Result of generated title using gpt-2 model

No	Keywords_6_Used	Original Title	Generated Title	Evaluation Metrics
1	['program', 'games', 'learns', 'chess', 'learning', 'play']	Learning to Play the Game of Chess	program games learns chess learning play a game of chess.The game is played by a group	Cosine = 0.5574 ROUGE-L = 0.4167 BERT_F1 = 0.8904 SciBERT_F1 = 0.6999
2	['security', 'privacy', 'computing', 'cloud', 'data', 'protection']	Data security and privacy in cloud computing	security privacy computing cloud data protection. The company has also announced that it will be launching a	Cosine = 0.3233 ROUGE-L = 0.1739 BERT_F1 = 0.8546 SciBERT_F1 = 0.6204
3	['processor', 'real-time', 'wavelet', 'decomposition', 'reconstruction', 'continuous']	Analog VLSI Processor Implementing the Continuous Wavelet Transform	processor real-time wavelet decomposition reconstruction continuous-wave wavefunction (CWR)	Cosine = 0.2038 ROUGE-L = 0.2222 BERT_F1 = 0.8644 SciBERT_F1 = 0.6590
4	['system', 'nonlinear', 'parametric', 'uncertainty', 'parameter', 'neural']	Neural Control for Nonlinear Dynamic Systems	system nonlinear parametric uncertainty parameter neural networks.The results of this study are presented in	Cosine = 0.1208 ROUGE-L = 0.0952 BERT_F1 = 0.8620 SciBERT_F1 = 0.5911

Tables 4, 5, and 6 show that our fine-tuned T5 model achieves better results than both KeyToText and GPT-2 in cosine similarity, ROUGE-L, BERT_F1, and SciBERT_F1. In Paper 1, the fine-tuned T5 model produces exactly the same title as the reference title, “Learning to Play the Game of Chess.” As a result, it reaches the maximum value for all metrics: cosine similarity = 1.0, ROUGE-L = 1.0, BERT_F1 = 1.0, and SciBERT_F1 = 1.0. In contrast, the title generated by KeyToText, “The program ‘games| learning|’ is ‘play.’”, achieves considerably lower scores. These values are lower (cosine = 0.3041, ROUGE-L = 0.3077, BERT_F1 = 0.8270, SciBERT_F1 = 0.5830), which shows that the title is different in both structure and meaning from the reference title. GPT-2 gives the title “program games learns chess learning play a game of chess. The game is played by a group,” which performs better than KeyToText (cosine = 0.5574, ROUGE-L = 0.4167, BERT_F1 = 0.8904, SciBERT_F1 = 0.6999). However, it still does not match the fine-tuned T5 model because the text is long and contains unnecessary phrases.

In Paper 2, “Data security and privacy in cloud computing,” the fine-tuned T5 model also provides strong results. It achieves a cosine similarity of 0.6328, a ROUGE-L score of 0.3077, a BERT_F1 of 0.9217, and a SciBERT_F1 of 0.8083 These values show that the model keeps the meaning of the reference title and follows its general structure. The generated title, “The Privacy-preserving cloud computing ,” is short and clearly expresses the main idea of the paper. In contrast, the KeyToText output is not close to the reference title. Its cosine similarity (0.0585) and ROUGE-L score (0.1176) are very low, which shows weak similarity in both wording and meaning. While the BERT-F1 score (0.8404) is reasonable, the SciBERT-F1 score (0.6185) indicates that the scientific terms are not well maintained. GPT-2 produces an output that contains some related keywords, but it is not written as a proper title and includes extra text. Its evaluation scores also show that the result is only partly related to the reference title and is not appropriate as a concise academic title. The same trend appears in papers 3 and 4. Overall, our model gives the best results compared with KeyToText and GPT-2. KeyToText often produces broken or unclear titles, whereas GPT-2 tends to generate longer titles than needed.

5. CONCLUSION

This study compared our model with two baseline methods, KeyToText and GPT-2, to examine their ability to generate titles for scientific papers. The results in Tables 4, 5, and 6 show that our model achieved the best performance across all four-evaluation metrics: cosine similarity, ROUGE-L, BERTF1, and SciBERTF1. This indicates that the model was more effective in preserving the meaning of the original title and maintaining its overall structure. Although the same keyword-based input was used for all three models, KeyToText often generated unclear titles or titles with grammatical problems, which is reflected in its lower cosine similarity, ROUGE-L, and SciBERT scores. GPT-2 performed better in terms of semantic similarity, but it often produced longer titles, making them less suitable as concise academic titles. Overall, the fine-tuned T5 model gave better results than both baseline methods in relevance, clarity, and consistency. This suggests that it is a reliable method for automatic title generation in scientific papers. The proposed framework was evaluated using the NIPS dataset in computer science and machine learning. In future work, the same pipeline can be applied to other domains and languages by developing domain-specific datasets, fine-tuning the model for each domain, and adjusting keyword weighting to produce accurate and fluent titles.

Conflicts of Interest

The authors declare no conflicts of interest.

REFERENCES

[1] A. Ganesan, B. Graf, J. Chen, and D. Roth, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1502.02367, 2015, doi:10.48550/arXiv.1412.3555.

[2] N. Sethi, P. Agrawal, V. Madaan, S. K. Singh, and A. Kumar, "Automated title generation in English language using NLP," Int. J. Control Theory Appl., vol. 9, no. 11, pp. 5159–5168, 2016.

[3] P. Biyani, K. Tsioutsiouliklis, and J. Blackmer, " '8 amazing secrets for getting more clicks': Detecting clickbaits in news streams using article informality," in Proc. AAAI Conf. Artificial Intelligence, vol. 30, no. 1, 2016, doi:10.1609/aaai.v30i1.9966.

[4] G. Erkan and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization," J. Artificial Intelligence Research, vol. 22, pp. 457–479, 2004, doi:10.1613/jair.1523.

[5] H. R. Jamali and M. Nikzad, "Article title type and its relation with the number of downloads and citations," Scientometrics, vol. 88, no. 2, pp. 653–661, 2011, doi:10.1007/s11192-011-0412-z.

[6] S.-C. Chen and L.-S. Lee, "Automatic title generation for Chinese spoken documents using an adaptive k-nearest neighbor approach," in Proc. EUROSPEECH 2003, Geneva, Switzerland, pp. 2813–2816, 2003, doi:10.21437/eurospeech.2003-749.

[7] S. Teufel, Argumentative Zoning: Information Extraction from Scientific Text, Ph.D. dissertation, Univ. of Edinburgh, Edinburgh, U.K., 1999.

[8] W. Li and J. Zhao, "TextRank algorithm by exploiting Wikipedia for short text keywords extraction," in Proc. 3rd Int. Conf. Information Science and Control Engineering (ICISCE), Beijing, China, pp. 683–686, 2016, doi:10.1109/ICISCE.2016.151.

[9] J. W. G. Putra and M. L. Khodra, "Automatic title generation in scientific articles for authorship assistance: A summarization approach," J. ICT Research & Applications, vol. 11, no. 3, pp. 253–267, 2017, doi:10.5614/itbj.ict.res.appl.2017.11.3.3.

[10] L. Alzubaidi, J. Zhang, A. J. Humaidi et al., "Review of deep learning: Concepts, CNN architectures, challenges, applications, and future directions," J. Big Data, vol. 8, art. no. 53, 2021, doi:10.1186/s40537-021-00444-8.

[11] L. Alzubaidi, J. Bai, A. Al-Sabaawi et al., "A survey on deep learning tools dealing with data scarcity: Definitions, challenges, solutions, tips, and applications," J. Big Data, vol. 10, art. no. 46, 2023, doi:10.1186/s40537-023-00727-2.

[12] B. Hamner, "NIPS papers," Kaggle Dataset, 2017, doi:10.34740/DVS/9097.

[13] A. Terko, E. Žunić, and D. Đonko, "NeurIPS conference papers classification based on topic modeling," in Proc. XXVII Int. Conf. Information, Communication and Automation Technologies (ICAT), Oct. 2019, pp. 1–5.

[14] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," Information, vol. 10, no. 4, 2019, doi:10.3390/info10040150.

[15] R. R. Tated and M. M. Ghonge, "A survey on text mining techniques and application," Int. J. Research in Advent Technology, pp. 380–385, 2015.

[16] B. Pahwa, S. Taruna, and N. Kasliwal, "Sentiment analysis-Strategy for text preprocessing," Int. J. Computer Applications, vol. 180, pp. 15–18, 2018, doi:10.5120/ijca2018916865.

[17] H. Saif, M. Fernandez, Y. He, and H. Alani, "On stopwords, filtering and data sparsity for sentiment analysis of Twitter," in Proc. 9th Int. Conf. Language Resources and Evaluation (LREC), pp. 810–817, 2014.

[18] H. P. Luhn, "The automatic creation of literature abstracts," IBM J. Research & Development, vol. 2, no. 2, pp. 159–165, 1958, doi:10.1147/rd.22.0159.

[19] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," J. Machine Learning Research, vol. 21, no. 140, pp. 1–67, 2020.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013, doi:10.48550/arXiv.1301.3781.

[21] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," arXiv preprint arXiv:1908.10084, 2019, doi:10.48550/arXiv.1908.10084.

[22] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in Proc. Workshop on Text Summarization Branches Out (WAS 2004), pp. 74–81, 2004.

[23] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," arXiv preprint arXiv:1904.09675, 2019, doi:10.48550/arXiv.1904.09675.

[24] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," arXiv preprint arXiv:1903.10676, 2019, doi:10.48550/arXiv.1903.10676.

[25] T. Rehman, D. K. Sanyal, and S. Chattopadhyay, "Can pre-trained language models generate titles for research papers?," arXiv preprint arXiv:2409.14602, 2024, doi:10.48550/arXiv.2409.14602.