



SUMOPE: Enhanced Hierarchical Summarization Model for Long Texts

Chao Chang^{1,2}, Junming Zhou¹, Xiangwei Zeng¹, and Yong Tang^{1,2}✉

¹ South China Normal University, Guangzhou 510631, Guangdong, China
{changchao,ytang}@m.scnu.edu.cn

² Pazhou Lab, Guangzhou 510330, Guangdong, China

Abstract. With the progress of the times, the ever-advancing and improving Internet technology and the ever-generating social media network platforms have made the amount of information in the network explode, which contains a massive scale of redundant content. Then how to quickly extract the key information from the huge amount of data becomes crucial. In this paper, we propose a novel enhanced hierarchical summarization model SUMOPE for long texts, which combines both extractive and abstractive methods to deal with long texts. Our model first uses an extractive method called SUMO to select key sentences from the long text and form a bridging document. Then, our model uses an abstractive method based on PEGASUS with a copy mechanism to generate the final summary from the bridging document. Our model can effectively capture the important information and relations in the long text and produce coherent and concise summaries. We evaluate our model on two datasets and show that it outperforms the state-of-the-art methods in terms of ROUGE scores and human evaluation.

Keywords: Text summarization · Transformer · Natural language processing

1 Introduction

Text summarization has been a well-established focus of research in the field of natural language processing, involving the creation of concise and coherent summaries for lengthy texts while preserving essential information. With the growing volume of online information, there is an increasing demand for efficient and accurate methods to summarize large amounts of textual data.

Currently, text summarization is approached through two primary methodologies: extractive and generative. Extractive summarization is based on statistical methods. It involves calculating the relevance of each sentence in the text based on certain extraction rules, such as keywords, position, and similarity to the overall text. It also involves selecting the top-ranked sentences as a summary. This method is relatively simple and has strong interpretability since the

extracted summary is faithful to the original text. However, extractive summarization depends largely on the quality of the source text. It may suffer from incoherent semantics and repetition in poorly structured text. Generative summarization methods can overcome these issues by not simply using the words and phrases from the source text to create the summary, but rather by extracting the meaning from the text and generating the summary one word at a time. Generative summarization is typically achieved through sequence-to-sequence models, but it may encounter problems such as out-of-vocabulary (OOV) and long-distance dependency issues. Furthermore, the encoding phase of summary generation can lead to a notable information loss due to the challenge of long-distance dependencies.

To address these challenges, we propose an enhanced hierarchical summarization model for long text, called SUMOPE. The first stage uses a hierarchical encoder-decoder architecture to extract salient sentences from the input text, and the second stage refines the selected sentences to produce a high-quality summary. Our model incorporates attention mechanisms and reinforcement learning to improve sentence selection and refinement. We evaluate SUMOPE on benchmark datasets and compare it with state-of-the-art models. Our experiments show that SUMOPE outperforms existing methods in automatic metrics and human evaluations.

The contributions of this paper can be summarized as follows:

- The paper proposes a novel enhanced hierarchical summarization model SUMOPE for long text, which addresses the challenge of generating high-quality summaries for lengthy texts.
- The enhanced hierarchical summarization model integrates both extractive and abstractive methods, leveraging the advantages of both to improve the quality of the generated summaries.
- The proposed model achieves state-of-the-art performance on two datasets, demonstrating its effectiveness and practicality for real-world applications.

2 Related Work

In the initial stages, extractive summarization methods were mostly unsupervised and based on statistics. These methods mainly relied on calculating the word frequency and the position of sentences to determine the score of each sentence in the text. Subsequently, they amalgamated the sentences with the most elevated scores to formulate a summary. Luhn, Jones et al. [1, 2] completed the task of text summarization by identifying keywords with significant information content in the text.

As research in machine learning and deep learning advances, supervised extractive summarization has become the mainstream research approach. In 2015, Can et al. [3] proposed a ranking framework for multi-document summarization. It uses Recursive Neural Networks to perform hierarchical regression and measure the salience of sentences and phrases in the parsing tree. The model learns ranking features automatically and concatenates them with hand-crafted

features of words to conduct hierarchical regressions. In 2017, Nallapati et al. [4] proposed an extractive summarization model based on Recurrent Neural Networks, which enables visualization of its predictions based on abstract features such as information content, salience, and novelty. Additionally, the model can be trained abtractively using human-generated reference summaries, eliminating the need for sentence-level extractive labels. In 2019, Liu et al. [5] presented a comprehensive framework for applying BERT, a pre-trained language model, to text summarization, covering both extractive and abtractive models. A document-level encoder based on BERT is introduced to capture the semantics of a document and obtain sentence embedding vector. For extractive summarization, inter-sentence transformer layers are stacked on top of the encoder. For abtractive summarization, a new fine-tuning schedule is proposed to handle the mismatch between the pre-trained encoder and the decoder. In 2021, Huang et al. [6] proposed an approach for extractive summarization that integrates discourse and coreference relationships by modeling the relations between text spans in a document using a heterogeneous graph. The graph contains three types of nodes, each corresponding to text spans of different granularity.

With further research into extractive summarization, researchers have discovered problems such as repetitive generation and lack of semantic coherence. In contrast, abtractive summarization, which generates new words and expressions based on the understanding of the text, is closer to human summarization thinking and emphasizes consistency and coherence [7]. In 2019, Dong et al. [8] proposed a comprehensive pre-trained language model capable of fine-tuned for both natural language comprehension and generation tasks. It utilizes a shared Transformer network along with specific self-attention masks to manage contextual information. In 2020, Zhang et al. [9] proposed a large Transformer-based encoder-decoder model that is pre-trained on massive text corpora with a new self-supervised objective tailored for abtractive text summarization. It generates summaries by removing/masking important sentences from the input document and generating them together as one output sequence from the remaining sentences. In 2020, Liu et al. [10] proposed a training paradigm for abtractive summarization models, which assumes a non-deterministic distribution to assign probability mass to different candidate summaries based on their quality.

While generative summarization models are capable of generating more accurate and readable summaries, they are limited by deep-learning techniques in obtaining text representations for long documents. In 2018, chen et al. [11] proposed a summarization model that follows a two-stage approach where salient sentences are selected and then rephrased to generate a concise summary. A sentence-level policy gradient method bridges the computation between the two neural networks while maintaining fluency. In 2021, Li et al. [12] proposed an extractive-abtractive approach to address the interpretability issue in abtractive summarization while avoiding the redundancy and lack of coherence in extractive summarization. The framework uses the Information Bottleneck principle to jointly train extraction and abstraction in an end-to-end fashion. It first extracts a pre-defined amount of evidence spans and then generates a summary

using only the evidence. In 2022, Xiong et al. [13] proposed a summarization model that uses elementary discourse units (EDUs) as the textual unit of content selection to generate high-quality summaries. The model first uses an EDU selector to choose salient content and then a generator model to rewrite the selected EDUs into the final summary. The group tag embedding is applied to determine the relevancy of each EDU in the entire document, allowing the generator to ingest the entire original document.

3 Proposed Technique

Inspired by SUMO [14] and PEGASUS [9], in this paper, we design a novel framework named SUMOPE to implement long text summarization, depicted in Fig. 1. Specifically, the extraction model based on SUMO extracts key sentences from long texts. These extracted sentences are then used as inputs to the generation model to produce the final summary. The transition document represents the set of extracted sentences from the extraction model. Its length falls between that of the original text and the summary, encompassing a significant portion of the crucial information found in the input document.

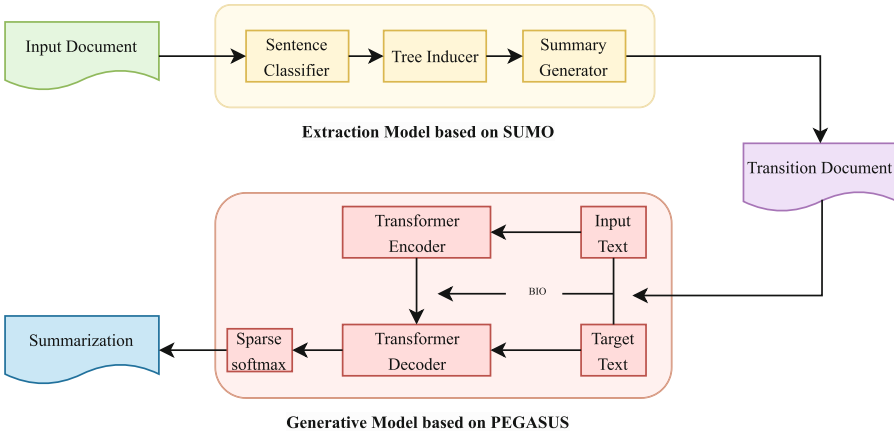


Fig. 1. The overview architecture of our proposed SUMOPE framework is with two modules: extraction model based on SUMO and generative model based on PEGASUS.

3.1 Extraction Model Based on SUMO

The extraction model based on SUMO is an approach to single-document summarization that uses tree induction to generate multi-root dependency trees that capture the connections between summary sentences and related content. This

technique hinges on the concept of framing extractive summarization as a tree induction challenge, where each root node within the tree symbolizes a summary sentence and the attached subtrees to it represent sentences whose content is related to and covered by the summary sentence.

The module comprises three main components: a sentence classifier, a tree inducer, and a summary generator. The sentence classifier uses a Transformer model with multi-head attention to classify each sentence in the input document as summary-worthy or not. The tree inducer then induces a multi-root dependency tree that captures the relationships between summary sentences and related content through an iterative refinement process that builds latent structures while using information learned in previous iterations. Finally, the summary generator selects the highest-scoring summary-worthy sentences from the induced tree and ensures that the selected sentences are coherent and cover all relevant aspects of the input document.

The SUMO algorithm generates these subtrees through iterative refinement and builds latent structures using information learned in previous iterations.

First, we decompose the input document D into individual sentences s_i . We then compute a score s_i^* for each sentence s_i , which reflects its importance for generating the summary. Precisely, we use the subsequent formula to calculate the score:

$$s_i^* = \sum_{j=1}^n w_j f_j(s_i) \quad (1)$$

where n is the number of features, w_j is the weight of feature j , and $f_j(s_i)$ is the value of feature j on sentence s_i .

Next, we select the highest-scoring sentence as a new root node and add all sentences dependent on it to form a new subtree. We then remove all sentences in this subtree from the document and add them to the summary set S . This process is repeated until all documents have been processed and all relevant subtrees have been added to S .

Finally, we use gradient descent to optimize feature weights and latent structures for generating more accurate, coherent, and diverse summaries. Specifically, we use a loss function L that balances coherence and diversity across documents and subtrees:

$$L = \sum_{i=1}^m \alpha_i L_i + \beta L_{div} \quad (2)$$

where α_i is the weight of document i , L_i is its loss function, β is a balancing factor, and L_{div} is the diversity loss function across all subtrees in S . We use the following formula to calculate L_{div} :

$$L_{div} = \sum_{i=1}^k \sum_{j=i+1}^k \frac{1}{d(T_i, T_j)} \quad (3)$$

where k is the number of subtrees in S , T_i and T_j are the i -th and j -th subtrees, and $d(T_i, T_j)$ is the distance between them.

One of the key advantages of this module is its ability to capture complex relationships between sentences in an input document. By inducing multi-root dependency trees, this approach can identify not only which sentences are most important for summarization but also how they relate to each other. This allows for more informative summaries that capture all important aspects of an input document while still being concise and easy to read.

3.2 Generative Model Based on PEGASUS

The generative model built upon the PEGASUS is a sequence-to-sequence architecture with gap-sentences generation as a pre-training objective tailored for abstractive text summarization. This technique involves the pre-training of expansive Transformer-based encoder-decoder models using extensive text datasets, all guided by a novel self-supervised objective.

The module architecture is based on a standard Transformer encoder-decoder. The encoder processes the input text, producing a sequence of hidden states that the decoder subsequently utilizes to produce the summary. The pre-training objective of PEGASUS involves generating gap-sentences, which are sentences that have been removed from the original text and replaced with special tokens. The module is trained to predict these gap-sentences given the surrounding context.

Formally, let $X = \{x_1, x_2, \dots, x_n\}$ be an input document consisting of n sentences, and let $Y = \{y_1, y_2, \dots, y_m\}$ be its corresponding summary consisting of m sentences. The goal of the PEGASUS algorithm is to learn a conditional probability distribution $p(Y|X)$ that generates a summary given an input document. This distribution can be factorized as follows:

$$p(Y|X) = \prod_{i=1}^m p(y_i | y_{<i}, X) \quad (4)$$

where $y_{<i}$ denotes the previously generated summary sentences.

To pre-train the model using gap-sentences generation, we first randomly select some sentences from the input document and replace them with special tokens. We then use this modified document as input and train the model to generate the missing sentences given the remaining context. The objective function used for pre-training is the negative log-likelihood of the ground-truth missing sentences:

$$\mathcal{L}_{pre} = - \sum_{i=1}^k \log p(y_i^* | y_{<i}, X) \quad (5)$$

where y_i^* denotes the ground-truth missing sentence and k is the number of missing sentences.

After pre-training, the model is fine-tuned on a specific summarization task using supervised learning. During fine-tuning, we use a similar objective function as in pre-training, but with the ground-truth summary sentences as targets:

$$\mathcal{L}_{fine} = - \sum_{i=1}^m \log p(y_i | y_{<i}, X) \quad (6)$$

where y_i denotes the ground-truth summary sentence.

The copy mechanism allows for direct copying of certain segments from the original text to the generated summary, thereby avoiding simple summarization. By preserving more original text information and avoiding information loss, especially for rare or non-existent words, the use of copy mechanism enhances the completeness and accuracy of the generated summary. In the decoder, a new label distribution is added for each token, as shown below:

$$p(y_t, z_t | y_{>t}, x) = p(y_t | y_{<t}, x) \cdot p(z_t | y_{<t}, x) \quad (7)$$

where B represents the token copied from the source text, I represents the token copied from the source text and forming a continuous segment with the previous tokens, and O represents the token not copied from the source text.

During the training phase, the model adds a sequence prediction task, and by calculating the longest common subsequence between the original text and the summary, corresponding BIO tags are obtained. During the prediction phase, for each step, the label Z_t is predicted first. If Z_t is O, no further processing is needed. If Z_t is B, it means that words that have never appeared in the original text need to be masked. If Z_t is I, it means that all corresponding n-grams unrelated to the original text need to be masked.

4 Experimental Evaluation

4.1 Experimental Setting

Data. In order to verify the effectiveness of our proposed approach, we conduct extensive experiments on two datasets, which are TTNews [15] and Scholat-News¹. As our model is designed for medium to long text, we controlled the length and clarity of the TTnews and SchoaltNews datasets by filtering out all articles with a length of less than 800 words. The datasets used in the experimentation are listed in Table 1.

Baselines

- LEAD is a classic method for text summarization that relies on the assumption that the first few sentences of a document contain the most important information. It involves selecting the first N sentences of a document as the summary, where N is a pre-defined number.

¹ <https://www.scholat.com/>.

Table 1. Data statistics.

Datasets		docs	avgArt	maxArt	minArt	avgSum	maxSum	minSum
TTNews	train	21359	1916	22312	800	42	78	21
	test	862	1909	17204	800	35	65	21
SchoaltNews	train	5988	3024	202089	800	41	122	2
	test	1216	1612	80831	800	45	98	2

- BertSum [5] is a text summarization approach that makes use of the Bidirectional Encoder Representations from Transformers (BERT) model. It introduces a document-level encoder rooted in BERT, capable of encoding an entire document and obtain deriving sentence representations. The extractive model is constructed atop this encoder by layering multiple inter-sentence Transformer layers, effectively capturing document-level attributes for sentence extraction.
- LongformerSum [16] is a text summarization method that leverages the Longformer model, which is designed to handle long sequences, for generating summaries of text documents.
- PGN [17] is a sequence-to-sequence framework that employs a soft attention distribution to generate an output sequence comprising elements sourced from the input document. PGN combines extractive and abstractive summarization methods by allowing the model to copy words directly from the source document while also generating new words to form a coherent summary.
- UniLM [8] is a pre-trained language model adaptable for tasks involving both comprehension and generation of natural language. The model is pre-trained via three distinct forms of language modeling tasks, making use of a common Transformer network alongside targeted self-attention masks, all strategically employed to regulate contextual understanding.
- BART [18] is a pre-training technique tailored for sequence-to-sequence models, seamlessly merging bidirectional and auto-regressive transformers. It uses a denoising autoencoder architecture, where text is corrupted with an arbitrary noising function and a sequence-to-sequence model is learned to reconstruct the original text.
- SUMO [14] is an extractive text summarization method that generates a summary by identifying key sentences in a document and organizing them into multiple subtrees. Each subtree consists of one or more root nodes, which are sentences relevant to the summary.
- PEGASUS [9] is a pre-training algorithm for abstractive text summarization that uses gap-sentences generation as a self-supervised objective. The model is trained to generate missing sentences given the remaining context, which allows it to capture the salient information in the input document. During fine-tuning, the model is optimized to generate a summary that captures the most important information in the input document.

Evaluations. ROUGE-N and ROUGE-L are two commonly used evaluation metrics for measuring the quality of text generation, such as machine translation, automatic summarization, question answering, and so on. They both compare the model-generated output with reference answers and calculate corresponding scores. In this paper, we use ROUGE-1, ROUGE-2, and ROUGE-L as evaluation metrics, where a higher score indicates higher quality of the generated text.

Environment and Parameter. The computer used in this study is equipped with an Intel(R) Xeon(R) Gold 5218 CPU @ 2.30 GHz and 256 GB of memory, with a Tesla V100 GPU with 32 GB of memory. Pycharm 2022 was used as the compiler, and Pytorch was adopted as the deep learning framework, with Python version 3.7. Experimental analysis and comparison were conducted using third-party libraries, including jieba2, bert4torch3, and Fengshenbang-LM4.

In the extraction model based on SUMO, the vocabulary size is 30,000, the maximum number of sentences is 200, the batch size is 256, the learning rate is 0.1, and the size of both EMD and hidden layers is 128. The transformer layers used to obtain sentence representations are set to 3, and the model is trained for 5 iterations.

In the generative model based on PEGASUS, we use the Chinese version of PEGASUS-BASE as the pre-trained model and adopt the Adam optimizer with a learning rate of $2e-5$. The batch size for training is 32, the epoch is 10, and the beam search width is 3. The maximum length of the generated summary is set to 90.

4.2 Result Analysis

As shown in Table 2 and Table 3, our model outperforms other models in terms of ROUGE-1, ROUGE-2, and ROUGE-L evaluation metrics on the TTnews and ScholatNews datasets.

Compared to the LEAD algorithm, BertSum demonstrates better performance, indicating that using Bert for summarization can greatly improve extraction accuracy. When comparing LongformerSum and BertSum, results show that replacing Bert with the Longformer model leads to improvement in evaluation metrics on two different long-text datasets. This is mainly due to the fact that Bert only retains the first 512 tokens, while Longformer can allow input with up to 4096 tokens, allowing the model to obtain more information. The results of LongformerSum and SUMO models demonstrate that treating extractive text summarization as a tree induction problem can produce results comparable to methods that use large-scale pre-trained models. Furthermore, the SUMO model outperforms LongformerSum in terms of training time and parameter size.

The performance of generative summarization models is excellent on text summarization tasks. Compared to BART and UniLM, the PEGASUS model has better performance, which demonstrates that specialized pre-training models may be more effective for specific tasks than general pre-training models. Therefore, the generation phase of our proposed model is optimized based on PEGASUS. Comparison of PEGASUS and our proposed model shows that SUMOPE

Table 2. Performance evaluation based on TTNews dataset.

Method	ROUGE-1	ROUGE-2	ROUGE-L
LEAD	21.63	7.2	16.97
BertSum	32.53	18.01	30.7
LongformerSum	37.45	21.76	31.91
PGN	30.91	16.88	28.47
UniLM	46.52	33.12	42.34
BART	54.8	39.14	51.33
SUMO	37.23	22.14	30.96
PEGASUS	55.54	41.34	52.81
SUMOPE	56.19	43.32	54.59

Table 3. Performance evaluation based on ScholatNews dataset.

Method	ROUGE-1	ROUGE-2	ROUGE-L
LEAD	37.27	26.44	32.67
BertSum	35.63	24.6	33.72
LongformerSum	38.33	27.4	32.73
PGN	36.36	28.75	32.89
UniLM	51.68	42.12	46.91
BART	60.59	54.05	57.92
SUMO	38.1	27.61	33.22
PEGASUS	63.23	58.37	62.06
SUMOPE	65.39	59.49	63.23

has improved to some extent in the three evaluation dimensions of Rouge-1, Rouge-2, and Rouge-L. Although most of the key information in the article is concentrated in the first 512 words, some critical information still exists in the second half of the article (contrasts and conclusions). Since our proposed model first preserves the key information of the article through extraction, the score of the generated summary will be higher.

Considering the limitations of solely using ROUGE metrics to evaluate the quality of generated summaries, as it cannot comprehensively assess whether the summary corresponds to the main theme of the article, this chapter adopts a subjective approach to evaluate the fidelity and fluency of the generated summaries. The study invites 20 students to subjectively evaluate the generated summaries based on fidelity and fluency, among other criteria. The participants are required to rate the summaries generated by different models for 9 randomly selected articles from the TTNEWS and ScholatNews test sets.

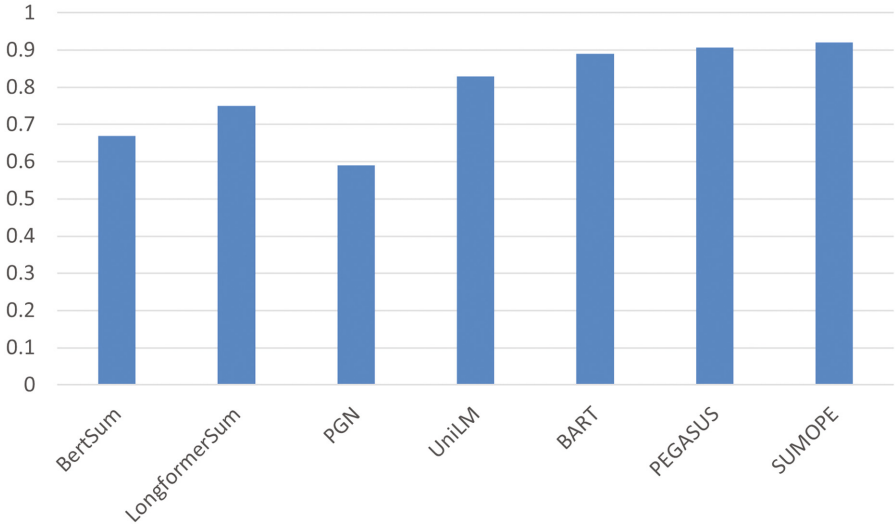


Fig. 2. The human evaluation results of the model are presented.

Based on the data presented in Fig. 2 it can be concluded that the summaries generated by the proposed model in the NLPCC and ScholatNews datasets are more in line with the main content of the articles, with a higher degree of matching with the standard summaries, more complete retention of key information, smoother semantic flow, and lower redundancy. Therefore, to some extent, it validates the effectiveness of the proposed model in generating high-quality summaries.

5 Conclusion

In this paper, we have proposed an enhanced hierarchical summarization model for long texts, SUMOPE, which combines extractive and abstractive methods to deal with long texts. The first stage of our model, called SUMO, selects key sentences from the input text to form a bridging document, and the second stage uses an abstractive method based on PEGASUS with a copy mechanism to generate the final summary. Our model has been evaluated on two datasets and has been shown to outperform the state-of-the-art methods in terms of ROUGE scores and human evaluation.

In future work, we plan to investigate the effectiveness of our model in other languages and domains. Additionally, we aim to explore the possibility of improving the performance of the model by incorporating other advanced techniques, such as reinforcement learning, to further enhance the selection and refinement of sentences. Overall, we believe that the proposed model has significant potential for improving the efficiency and accuracy of text summarization in various applications.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Grant U1811263.

References

1. Luhn, H.P.: The automatic creation of literature abstracts. *IBM J. Res. Dev.* **2**(2), 159–165 (1958). <https://doi.org/10.1147/rd.22.0159>
2. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. *J. Documentation* **60**(5), 493–502 (2004)
3. Cao, Z., Wei, F., Dong, L., Li, S., Zhou, M.: Ranking with recursive neural networks and its application to multi-document summarization. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 25–30 January 2015, Austin, Texas, USA, pp. 2153–2159. AAAI Press (2015)
4. Nallapati, R., Zhai, F., Zhou, B.: SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 4–9 February 2017, San Francisco, California, USA, pp. 3075–3081. AAAI Press (2017)
5. Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, Hong Kong, China, 3–7 November 2019, pp. 3728–3738. Association for Computational Linguistics (2019)
6. Huang, Y.J., Kurohashi, S.: Extractive summarization considering discourse and coreference relations based on heterogeneous graph. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021*, Online, 19–23 April 2021, pp. 3046–3052. Association for Computational Linguistics (2021)
7. Ma, C., Zhang, W.E., Guo, M., Wang, H., Sheng, Q.Z.: Multi-document summarization via deep learning techniques: a survey. *ACM Comput. Surv.* **55**(5), 102:1–102:37 (2023). <https://doi.org/10.1145/3529754>
8. Dong, L., et al.: Unified language model pre-training for natural language understanding and generation. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, Vancouver, BC, Canada, 8–14 December 2019, pp. 13042–13054 (2019)
9. Zhang, J., Zhao, Y., Saleh, M., Liu, P.J.: PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, 13–18 July 2020, Virtual Event. *Proceedings of Machine Learning Research*, vol. 119, pp. 11328–11339. PMLR (2020)
10. Liu, Y., Liu, P., Radev, D.R., Neubig, G.: BRIO: bringing order to abstractive summarization. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022*, Dublin, Ireland, 22–27 May 2022, pp. 2890–2903. Association for Computational Linguistics (2022)
11. Chen, Y., Bansal, M.: Fast abstractive summarization with reinforce-selected sentence rewriting. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, Melbourne, Australia, 15–20 July 2018, Volume 1: Long Papers, pp. 675–686. Association for Computational Linguistics (2018)

12. Li, H., et al.: EASE: extractive-abstractive summarization end-to-end using the information bottleneck principle. In: Proceedings of the Third Workshop on New Frontiers in Summarization, pp. 85–95 (2021)
13. Xiong, Y., Racharak, T., Nguyen, M.L.: Extractive elementary discourse units for improving abstractive summarization. In: SIGIR 2022: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022, pp. 2675–2679. ACM (2022)
14. Liu, Y., Titov, I., Lapata, M.: Single document summarization as tree induction. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019, Volume 1 (Long and Short Papers), pp. 1745–1755. Association for Computational Linguistics (2019)
15. Hua, L., Wan, X., Li, L.: Overview of the NLPCC 2017 shared task: single document summarization. In: Huang, X., Jiang, J., Zhao, D., Feng, Y., Hong, Yu. (eds.) NLPCC 2017. LNCS (LNAI), vol. 10619, pp. 942–947. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73618-1_84
16. Wei, F., Yang, J., Mao, Q., Qin, H., Dabrowski, A.: An empirical comparison of distilBERT, longformer and logistic regression for predictive coding. In: IEEE International Conference on Big Data, Big Data 2022, Osaka, Japan, 17–20 December 2022, pp. 3336–3340. IEEE (2022)
17. See, A., Liu, P.J., Manning, C.D.: Get to the point: summarization with pointer-generator networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, 30 July–4 August, Volume 1: Long Papers, pp. 1073–1083. Association for Computational Linguistics (2017)
18. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880 (2020)