Array 20 (2023) 100322

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/locate/array

Extractive social media text summarization based on MFMMR-BertSum

Junqing Fan [a], Xiaorong Tian [a], Chengyao Lv [b,*], Simin Zhang [a], Yuewei Wang [a], Junfeng Zhang [b]

[a] School of Computer Science, China University of Geosciences, Wuhan, 430074, China
[b] School of Foreign Language, China University of Geosciences, Wuhan, 430074, China

ARTICLE INFO

Keywords:
Natural language processing
Abstractive text summarization
BERT
Machine learning

ABSTRACT

The advancement of computer technology has led to an overwhelming amount of textual information, hindering the efficiency of knowledge intake. To address this issue, various text summarization techniques have been developed, including statistics, graph sorting, machine learning, and deep learning. However, the rich semantic features of text often interfere with the abstract effects and lack effective processing of redundant information. In this paper, we propose the Multi-Features Maximal Marginal Relevance BERT (MFMMR-BertSum) model for Extractive Summarization, which utilizes the pre-trained model BERT to tackle the text summarization task. The model incorporates a classification layer for extractive summarization. Additionally, the Maximal Marginal Relevance (MMR) component is utilized to remove information redundancy and optimize the summary results. The proposed method outperforms other sentence-level extractive summarization baseline methods on the CNN/DailyMail dataset, thus verifying its effectiveness.

1. Introduction

In the era of big data, the rapid development of social media constantly supplies a bulk of information. Text content, as a dominant medium in social media, is an efficient approach to conveying real-time news and opinions. However, the abundance of descriptions and interpretations often obscures the concrete opinion in a content, hindering the timely acquisition of vital information. Text summarization is a technique used to condense long text into a shorter abstract while retaining its original meaning [1]. Automatically generating key information from massive text can significantly improve efficiency compared to traditional manual summarization [2]. Automatic text summarization methods can be divided into two categories based on the relationship between the abstract and the original text: extractive summarization and abstractive summarization [3]. Extractive summarization extracts keywords from the source document to form a summary. However, this approach may result in a final summary that lacks coherence between sentences and may contain redundant information. While abstractive summarization generates new words to form a summary based on the content of the source document.

The mainstream approaches in the field of automatic text summarization include those based on statistics, graph sorting [4], machine learning, and deep learning. Statistical-based text summarization methods are simple and intuitive, solely considering word-surface features, while ignoring the grasp of word-sense relationships.[5] Methods based on graph sorting are suitable for loosely structured texts [6], but they do not take contextual information into account. The development of deep learning techniques has facilitated breakthroughs in natural language processing. The BERT model [7] is a pre-trained model that has been trained on large-scale datasets, demonstrating powerful generalization capabilities. BERT uses word-level inputs while extractive summarization is a sentence-level task. Therefore it is impossible to fine-tune the BERT pre-trained model directly for automatic text summarization tasks.

To address the issue, we propose the MFMMR-BertSum model. The primary concept of this method is to incorporate the pre-trained BERT model into the social media text summarization task, modifying its input representation to capture sentence features and differentiate them. Subsequently, a classification layer is constructed after the model's output to extract the summarized sentences, enabling its application to the text summarization task. Additionally, a de-redundancy component is added to further optimize the summarization results based on the principles of MMR [8]. The key innovations of this model are as follows:

- Modify the input representation of BERT to sentence level, and add a classification layer after its output, so that the model can be applied to extractive summarization tasks. The combination of different pre-trained models and classification layers is designed to obtain the optimal social media text summarization model.
- MMR is utilized to add a component to remove redundancy for the prediction process of the model. In the feature extraction

* Corresponding author.
E-mail addresses: fanjq@cug.edu.cn (J. Fan), xiaorongtian@cug.edu.cn (X. Tian), 83520583@qq.com (C. Lv), simin.z.cn@icloud.com (S. Zhang), yuewei.w@cug.edu.cn (Y. Wang), zjfeng@cug.edu.cn (J. Zhang).

https://doi.org/10.1016/j.array.2023.100322
Received 26 July 2023; Received in revised form 14 September 2023; Accepted 30 September 2023
Available online 4 October 2023

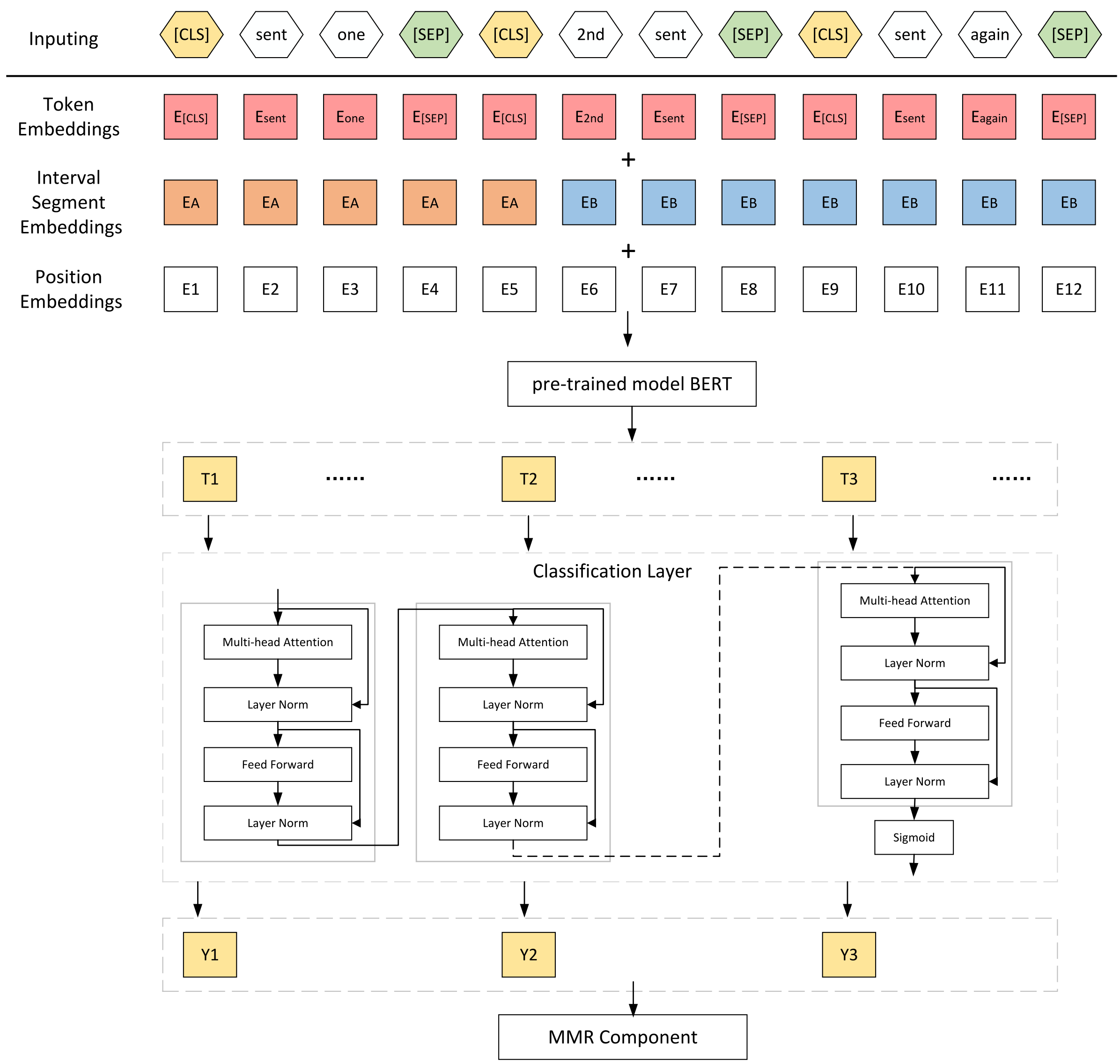

Fig. 1. The overall architecture of MFMMR-BertSum Model.

process, the weighted combination of multiple features is used as the final feature. The process of model extraction summary is also improved, and a temporary summary set is used to reduce the time complexity of the method and optimize the summary results.

This paper will be developed from the following aspects: the second part focuses on the work related to text summarization; the third part improves the BERT language model to implement the MFMMR-BertSum model applied to the social media text summarization work; and finally, the work of this paper is organized and summarized.

2. Related work

Extractive summarization was the dominant approach to automatic text summarization before the advent of deep learning technology [9]. The basic principle behind statistical methods for extractive summarization involves analyzing word frequency, sentence position, and their weighted combinations to determine sentence importance. According to Luhn, certain words may have greater significance if they appear more frequently in the text. Baxendale found in his research on the summary of sentence position features that it has a strong correlation with the text topic. Edmundson pointed out that some specific words are related to the importance of the sentence. Optimization-based methods [10] usually formalize text summarization as a mathematical problem with constraints. MMR [8] algorithm is one of the classical methods. It takes the linear combination of the similarity of documents with respect to the query and the similarity with documents that have been previously selected for summarization as the "edge correlation", and maximizes this edge correlation value during the retrieval and summarization process to gradually obtain the final summary. Cheng [11] ranked each sentence based on the probability that it would become a summary using SVM. The final summary set is obtained by using the improved MMR algorithm to select the sentences at the edge of the summary ratio.

The emergence of deep learning technology has revolutionized the field of extractive text summarization, bringing about significant progress and advancements. Liu [12] applied deep learning to the field of text summarization for the first time and proposed a text summarization method based on RBM. The emergence of pre-trained models like BERT [7] has brought natural language processing into a new era. BERT uses the encoder part of the Transformer as the main framework of the model. Through the joint adjustment of the context of each layer to predict the deep bidirectional representation, capturing the bidirectional context relationship in the statement. The BertSum model proposed by Liu [13] is the first BERT-based text summarization model. Some modifications have been made to the embedding of the BERT model for the purpose of extractive summarization. Yuan [14] added the hierarchical graph mask to BERT to make full use of the structural information between different semantic levels and extract the semantic units at the fact level to obtain a better summary. Srikanth [15] used the K-means algorithm to cluster the sentence representations output by the BERT model and introduced a dynamic method to determine the appropriate number of sentences from the cluster. Ma [16] propose a topic-aware extractive and abstractive summarization model based on bidirectional encoder representations from Transformer, which focuses on pre-trained external knowledge and topic mining to capture more accurate contextual representations. Kieuvongngam [17] used BERT and GPT-2 to extract and generate abstract in COVID-19 medical papers and analyzed the differences and improvement strategies of the two methods.

Despite the effectiveness of Bert-based extractive summarization methods, there is still room for improvement. The current absence of efficient methods for the redundancy handling of summary content is one of the areas that can be addressed.

3. MFMMR-BertSum

The pre-trained model BERT boasts semantically-rich representation capabilities that effectively address challenges such as polysemy and long-distance dependencies in natural language processing. The input representation of the BERT model consists of token embeddings, segment embeddings, and position embeddings. The body of the model is a multi-layer bidirectional transformer structure and the output can be trained on downstream tasks through fine-tuning by connecting it to neural networks.

In this section, we combine the features of BERT model and the requirements of extractive text summarization tasks to construct and propose the MFMMR-BertSum model. As shown in Fig. 1. To create our extractive summarization model, we begin by feeding the text content into a modified pre-trained BERT model, which captures sentence features through tag-based differentiation. This step yields the sentence vector code. The output is subjected to a combination of linear classification and Transformer classification at the classification layer. This process produces a prediction score, which is then sorted in descending order to generate a preliminary summary. To refine the summary further, we perform a second screening step designed to eliminate redundant components. The resulting summary is our final output.

3.1. Input representation

Since BERT is trained based on word embeddings, it can neither directly obtain sentence representations nor distinguish between multiple sentences. To address the above problems, we refer to the study of Liu [13] to make some modifications to the input of BERT, and the modified model is called BertSum.

To enable the BERT model can be trained on the extractive social media text summarization task, we add the [CLS] tag before each sentence of the input and keep the [SEP] tag after each sentence. The token Embeddings layer converts each word into a vector. At the part of the segment embeddings, assigning the value E_A or E_B to $sent_i$ according to the parity of the sentence number, so that it can differentiate the input of multiple sentences. Additionally, position embeddings are employed to capture the word sequence. After modification, the vector corresponding to each [CLS] tag is the sentence feature captured by the model.

3.2. Classification layer

The classification layer is built to train the sentence features to determine the importance of the sentence under the document-level features after the BertSum model has been used to acquire the features of the sentence.

3.2.1. Linear classifier

After the BertSum output, add one or more linear layers and then apply the Sigmoid function to obtain the final predicted values. The vector T_i is the ith [CLS] symbol from BertSum that will be used as the representation of the $sent_i$. For each sentence, the probability of it being used as a summary through the classification layer is calculated as $\hat{Y}_i$.

$$\hat{Y}_i = \sigma(w_o T_i + b_o) \tag{1}$$

Where σ represents the Sigmoid function, w_o and b_o are the weight and deviation.

3.2.2. Transformer classifier

Transformer is a framework based on self-attention mechanism. The calculation method is shown in Eqs. (2), (3):

$$\widetilde{h}^l = LN(h^{l-1} + MHAt(h^{l-1})) \tag{2}$$

$$h^l = LN(\widetilde{h}^l + FFN(\widetilde{h}^l)) \tag{3}$$

Among them, $h^0 = PosEmb(T)$, T is the sentence vector output by the BertSum model, and $PosEmb(T)$ represents the position embeddings for vector T. The superscript l represents the depth of the stacked layer.

The layer normalization procedure (LN) is used to normalize all neurons in a sample's same layer. $MHAtt$ is the multi-head attention operation. FFN is the feed forward network of Transformer.

Finally, the Sigmoid function is added to the output of the Transformer to realize the classification. The vector calculation method of the predicted value $\hat{Y}_i$ is shown in Eq. (4):

$$\hat{Y}_i = \sigma(w_o h_i^L + b_o) \tag{4}$$

where h^L is the vector for $sent_i$ from the Lth layer of the Transformer. σ represents the Sigmoid function, w_o and b_o are the weight and deviation.

3.3. MMR-based component

To reduce redundancy in the abstract, we have incorporated an MMR component into the prediction phase of our model. Traditional MMR only considers word-level features and disregards other aspects, hindering the quality of the final summary. Moreover, the time complexity of MMR based on greedy selection depends on the number of summary sentences. Directly applying the original MMR could result in a large increase in the model's processing time. Taking on these difficulties, we propose an MFMMR algorithm that utilizes a weighted combination of multiple features as sentence features during the feature extraction process. Additionally, we improve the summary model extraction process by implementing a temporary summary set to reduce the time complexity of the method. By using these techniques, we can generate a more concise and accurate summary while streamlining the computation of the model.

3.3.1. MFMMR

The MFMMR algorithm uses a weighted combination of multiple features as sentence features in the feature extraction process.

1 TF-IDF

The frequencies of different keywords in the sentence are recorded as the score of the sentence.

$$K_i = Count(keywords \in S_i) \tag{5}$$

where S_i represents the ith sentence. 2 Sentence Position and Numerical Information

Typically, the opening sentence of a document serves as a potential candidate for the final summary. In light of the sentence's placement, Eq. (6) is used to determine the sentence's weight.

$$L_i = \begin{cases} 1 - \frac{i}{3} & 1 \le i \le 3 \\ 0 & others \end{cases} \tag{6}$$

In addition, sentences containing numbers usually indicate some key information.

$$N_i = \begin{cases} 1 & \textit{Sentence contains number} \\ 0 & \textit{Sentence does not contain number} \end{cases} \tag{7}$$

The value after the average of the two features is considered one feature, written as Eq. (8):

$$F_i = \frac{1}{2}\left(L_i + N_i\right) \tag{8}$$

3 Word2vec

Word2vec is used to vectorize the sentence. A sentence vector is Si, and the average value of other sentence vectors in the document is used as the vector value of the document Dj. The similarity between the sentence and the document is shown in Eq. (9):

$$W_i = sim\left(S_i, D_j\right) = \frac{S_i \cdot D_j}{\|S_i\| \times \|D_j\|} \tag{9}$$

4 Emotional Characteristic Value

The subjectivity of sentences can be calculated through simple sentiment analysis [18]. Define the emotional value of a sentence as below:

$$E_i = 1 - subjectivity(S_i) \tag{10}$$

where S_i represents the sentence, and $subjectivity(S_i)$ represents the subjective score of the sentence.

The weighted combination of the above features is used as the final sentence score, as shown in Eq. (11):

$$Score_i = \alpha K_i + \beta F_i + \gamma W_i + \delta E_i \tag{11}$$

Where α, β, γ and δ are weighted coefficients, satisfying $\alpha+\beta+\gamma+\delta = 1$. In practice, the weighted coefficients can be flexibly adjusted according to the importance of different features and the range of values.

Combined with the actual needs, this paper designs its MFMMR algorithm formula for the extractive summarization task as shown in Eq. (12):

$$MFMMR(C_i) = \lambda \cdot Score_i - (1-\lambda)\max_{S_j \in S}(Sim(C_i, S_j)) \tag{12}$$

C_i is the candidate sentence to be classified in the document, S represents the summary sentence set, and S_j is the sentence that has been selected as the summary. The basic idea is to use the hyperparameter λ to penalize the sentence scores that are too similar to the summary sentence to reduce the summary redundancy.

For the weighted coefficients α, β, γ and δ used in this paper, α, β, and γ are set to take 0.15 to 0.35 in 0.05 increment, δ=1-$\alpha - \beta - \gamma$ and δ> 0. Where α=0.25, β=0.2, γ=0.2, and δ=0.35, ROUGE-1 and ROUGE-L achieve the maximum value, ROUGE-2 is nearly equal to the maximum value. At this time, the sentence score can fully take into account the contribution of multiple features.

To investigate the effect of the hyperparameter λ and the word vector dimension on the summary performance. λ is set to take 0.5 to 0.9 in 0.1 increment, and the word vector dimensions are taken to be 100 to 300 in 50 increments. ROUGE-1, ROUGE-2 and ROUGE-L are used as evaluation metrics. The result indicated when λ takes the value of 0.8, ROUGE-1 and ROUGE-L take the maximum value, and ROUGE-2 is also basically close to the maximum value. At this time, the sentence score term and redundancy term weights are best assigned. Furthermore, the ROUGE score changes very little under different word vector dimensions. Considering that a word vector with too large a dimension increases the complexity of the model and thus the running time, it is more appropriate when the word vector dimension is taken as 100. Through this experimentation, we determined that the MFMMR algorithm yielded the best summary performance with a hyperparameter λ of 0.8 and a word vector dimension of 100. These parameters were utilized for subsequent calculations.

3.3.2. Workflow of MMR-based component

As shown in algorithm 1, the main concept behind this component is to utilize a temporary summary set during the process of extracting summaries. Initially, the sentences selected as summaries are added to the temporary summary set. Then, the MFMMR between subsequent candidate sentences and the temporary summary set is calculated. If the maximum edge correlation falls below a certain threshold, the candidate sentence is deemed less relevant or redundant and subsequently discarded. On the other hand, if the maximum edge correlation surpasses the threshold, the candidate sentence is added to the temporary summary set. Once the number of sentences in the temporary summary set reaches a specific level, those sentences are pushed to the final summary set. The time complexity depends on the size of the temporary summary set, allowing for manageable time consumption, because of which it is suited for deep learning-based methods.

Algorithm 1: MMR Component

```
Data: Sentence vector T_s, temporary summary set Temp_s
/* T_s is corresponding to the candidate summary
   sentence output by the classification layer
   */
Result: final summary set Final_s
for each T_i ∈ T_s do
    if Temp_s == ∅ then
        goto Flag
    end
    else
        calculate MFMMR of T_i and Temp_s
        if MFMMR > threshold then
            continue
        end
        else
            goto Flag
        end
    end
    Flag:
    Temp_s = Temp_s+T_i
    if Temp_s if full then
        Final_s=Final_s+Temp_s
        Temp_s=∅
        if Final_s if full then
            return Final_s
        end
        else
            continue
        end
    end
    else
        continue
    end
end
```

4. Experiment

4.1. Experimental setup

Firstly, the MFMMR algorithm is employed for extractive summarization tasks alongside baseline algorithms including Lead, SumBasic [19], TextRank [20], LexRank [21], and MMR. Additionally, the number of sentences in the final summary is restricted to either one or three, allowing for an assessment of the effectiveness of the MFMMR algorithm in both single-sentence and multi-sentence summarization tasks.

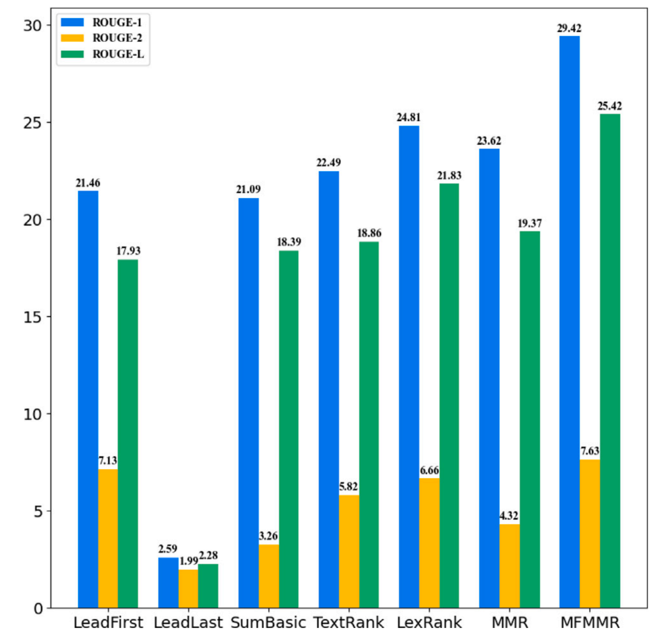

Fig. 2. Single-Sentence Summary Performance of MFMMR and Baseline Methods.

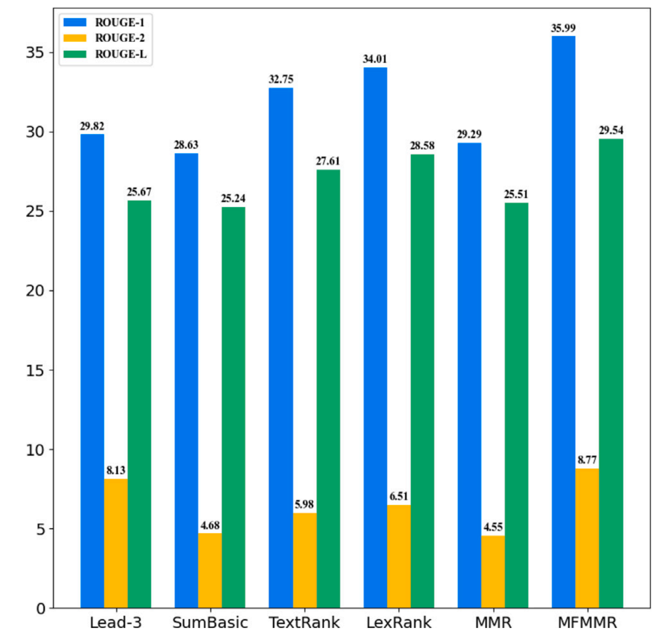

Fig. 3. Multi-sentence Summary Performance of MFMMR and Baseline Methods.

Next, to investigate the impact of different classification layers and pre-trained models, five pre-trained models, including Bert, DistilBert [22], RoBERTa [23], DistilRoBERTa, and MPNet [24]. And six classification layers, including 1∼2-layer linear classifier (L1, L2), 1∼3-layer Transformer classifier (T1, T2, T3), double-layer Transformer plus single-layer linear classifier (T2L) were used to compare the combined summary effects.

Lastly, the MFMMR-BertSum algorithm is used to compare with Lead-3, SummaRuNNer [25], LANTENT [26], REFRESH [27], Bert-WCSS [28], NEUSUM [29], PGN [30], BottomUp [31], DCA [32], SummaRuNNer-PGN [33], and PreSum [34]. Bert and the classification layer are jointly trained, dropout is set to 0.1, and the optimizer uses Adam. The parameters $\beta_1 = 0.9$, $\beta_2 = 0.9$, the learning rate adopts the default value of Transformer, and the maximum epoch of model training is set to 3. Select the first three sentences as a summary. The evaluation indexes used in the experiment are still ROUGE-1, ROUGE-2, and ROUGE-L [35].

4.2. Dataset

The data set used in this paper is CNN/DailyMail [36]. In the data preprocessing stage, a greedy algorithm is used to generate a prediction summary for each document. The label value of the predicate summary sentence is set to 1, others are set to 0, and the obtained sequence is used as the label in the training process.

4.3. Experimental result

As shown in Figs. 2,3, whether it is a single-sentence or a multi-sentence summary, the MFMMR algorithm considers multiple features as the calculation method of sentence score. On the basis of MMR, it greatly eliminates the negative impact of single feature, and obtains the highest score in baseline methods, verifying the algorithm's effectiveness in extractive social media text summarization.

Among the various combinations of pre-trained models and classification layers in Table 1, the model utilizing DistilBert with the three-layer Transformer classifier exhibited superior performance compared to other combinations. It performed 42.67% in ROUGE-1, 19.63% in ROUGE-2 and 39.08% in ROUGE-L. The model summary was further optimized by adding the MMR redundancy component after its output, and the metrics were improved by 0.07%, 0.22%, and 0.11%. This optimal combination was used for subsequent experiments (see Table 2).

Table 3 demonstrates the significant improvement achieved by MFMMR-BertSum over the baseline approach, positioning it as the top-performing model among both extractive and abstractive models. Among the baseline methods, the best results in the sentence-level-based extractive summary come from NEUSUM, where our model is 1.15%, 0.84%, and 1.21% higher than the ROUGE-1, ROUGE-2, and ROUGE-L of this model. These results indicate a substantial enhancement in the alignment with reference summaries across different sliding window scales, word order, and sentence structure, thus validating the effectiveness of the proposed MFMMR-BertSum model. Meanwhile, the hybrid model SummaRuNNer-PGN achieves better results than the separate one. Accordingly, we speculate that hybrid models combining the advantages of extractive and abstractive summarization have plenty of potential for growth.

5. Conclusion and future work

In this research paper, we propose the MFMMR algorithm, considering multiple features in the conventional MMR sentence scoring process. Significantly mitigates the adverse effects of relying on single feature for calculating sentence scores. The MFMMR-BertSum model is proposed by modifying the input representation of Bert and adding a classification layer with MMR components to reduce the redundancy problem in extractive summarization. Tests were conducted on the CNN/DailyMail dataset, and the results indicate MMR-BertSum has significant improvements compared to the baseline approach on ROUGE-1, ROUGE-2, and ROUGE-L. In the meanwhile, the hybrid model of extractive and abstractive achieves better results than using both models alone. An attempt can be made to combine the advantages of both methods to improve the overall performance of the summaries.

Table 1
Combination effect of different pre-trained models and classification layers.

Pre-traine model	Evaluation indexes	Classification layers					
		L1	L2	T1	T2	T2L	T3
Bert-base-uncased	ROUGE-1	41.91	42.01	41.92	42.1	42.05	41.91
	ROUGE-2	19.07	19.15	19	19.13	19.2	19.05
	ROUGE-L	38.43	38.49	38.44	38.58	38.57	38.42
Distilbert-base-uncased	ROUGE-1	42.36	42.41	42.62	42.6	42.46	42.67
	ROUGE-2	19.40	19.48	19.63	19.61	19.47	19.63
	ROUGE-L	38.80	38.86	39.03	39.01	38.9	39.08
Roberta-base	ROUGE-1	42.45	42.53	42.23	42.55	41.99	42.32
	ROUGE-2	19.39	19.49	19.21	19.45	19.02	19.29
	ROUGE-L	38.84	38.94	38.68	38.94	38.41	38.71
Distilroberta-base	ROUGE-1	42.24	42.15	42.43	42.30	41.93	42.32
	ROUGE-2	19.24	19.18	19.41	19.30	18.97	19.30
	ROUGE-L	38.63	38.53	38.91	38.67	38.30	38.72
Mpnet-base	ROUGE-1	42.10	42.24	42.27	42.16	42.08	42.13
	ROUGE-2	19.05	19.18	19.21	19.09	19.02	19.04
	ROUGE-L	38.49	38.62	38.69	38.57	38.47	38.51

Table 2
Effects of MMR components.

	ROUGE-1	ROUGE-2	ROUGE-L
With MMR	42.74	19.85	39.19
Without MMR	42.67	19.63	39.08

Table 3
Comparison of the performance of different models.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Lead-3	40.34	17.70	36.57
SummaRuNNer	39.60	16.20	35.30
LANTENT	41.16	15.75	39.08
REFRESH	40.00	18.20	36.60
Bert-WCSS	41.40	17.90	37.90
NEUSUM	41.59	19.01	37.98
PGN	39.53	17.28	36.38
BottomUp	41.22	18.68	38.34
DCA	41.69	19.47	37.92
SummaRuNNer-PGN	40.68	17.97	37.13
PreSumm	42.13	19.60	39.18
MFMMR-BertSum	42.74	19.85	39.19

CRediT authorship contribution statement

Junqing Fan: Conceptualization, Methodology. **Xiaorong Tian:** Writing – original draft, Visualization, Investigation. **Chengyao Lv:** Supervision, Funding acquisition. **Simin Zhang:** Investigation, Methodology, Writing – original draft. **Yuewei Wang:** Investigation, Editing. **Junfeng Zhang:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This paper is partially supported by the National Natural Science Foundation of China (No. 62076224).

References

[1] Radev DR, Hovy E, McKeown K. Introduction to the special issue on summarization. Comput Linguist 2002;28(4):399–408.
[2] Liu J, Chen Y, Huang X, Li J, Min G. GNN-based long and short term preference modeling for next-location prediction. Inform Sci 2023;629:1–14.
[3] El-Kassas WS, Salama CR, Rafea AA, Mohamed HK. Automatic text summarization: A comprehensive survey. Expert Syst Appl 2021;165:113679.
[4] Jia Y, Gu Z, Jiang Z, Gao C, Yang J. Persistent graph stream summarization for real-time graph analytics. World Wide Web 2023;1–21.
[5] Haldar NAH, Li J, Ali ME, Cai T, Chen Y, Sellis T, et al. Top-k socio-spatial co-engaged location selection for social users. IEEE Trans Knowl Data Eng 2023;35(5):5325–40.
[6] Bahani M, El Ouaazizi A, Maalmi K. AraBERT and DF-GAN fusion for Arabic text-to-image generation. Array 2022;16:100260.
[7] Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. 2019, arXiv:1810.04805.
[8] Carbonell J, Goldstein J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval. 1998, p. 335–6.
[9] Esposito M, Fujita H, Minutolo A, Pota M. Deep learning for natural language processing: Emerging methods and applications. Array 2022;14:100138.
[10] Jain D, Borah MD, Biswas A. Fine-tuning textrank for legal document summarization: A Bayesian optimization based approach. In: Forum for information retrieval evaluation. 2020, p. 41–8.
[11] Kun C, Chuanyi L, Xinxin J, Jidong G, Bin L. News summarization extracting method based on improved MMR algorithm. J Appl Sci 2021;39(3):442–3.
[12] Liu Y, Zhong S-h, Li W. Query-oriented multi-document summarization via unsupervised deep learning. In: Proceedings of the AAAI conference on artificial intelligence, vol. 26, no. 1. 2012, p. 1699–705.
[13] Liu Y. Fine-tune BERT for extractive summarization. 2019, arXiv:1903.10318.
[14] Yuan R, Wang Z, Li W. Fact-level extractive summarization with hierarchical graph mask on BERT. 2020, arXiv:2011.09739.
[15] Srikanth A, Umasankar AS, Thanu S, Nirmala SJ. Extractive text summarization using dynamic clustering and co-reference on BERT. In: 2020 5th international conference on computing, communication and security. IEEE; 2020, p. 1–5.
[16] Ma T, Pan Q, Rong H, Qian Y, Tian Y, Al-Nabhan N. T-bertsum: Topic-aware text summarization based on bert. IEEE Trans Comput Soc Syst 2021;9(3):879–90.
[17] Kieuvongngam V, Tan B, Niu Y. Automatic text summarization of COVID-19 medical research articles using BERT and GPT-2. 2020, arXiv:2006.01997.
[18] Bhowmik NR, Arifuzzaman M, Mondal MRH. Sentiment analysis on Bangla text using extended lexicon dictionary and deep learning algorithms. Array 2022;13:100123.
[19] Nenkova A, Vanderwende L. The impact of frequency on summarization, vol. 101. Tech. rep. MSR-TR-2005, Redmond, Washington: Microsoft Research; 2005.
[20] Mihalcea R, Tarau P. TextRank: Bringing order into text. Barcelona, Spain: Association for Computational Linguistics; 2004, p. 404–11.
[21] Erkan G, Radev DR. LexRank: Graph-based lexical centrality as salience in text summarization. J Artif Int Res 2004;22(1):457–79.
[22] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2020.

[23] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. RoBERTa: A robustly optimized BERT pretraining approach. 2019, arXiv:1907.11692.
[24] Song K, Tan X, Qin T, Lu J, Liu T-Y. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H, editors. MPNet: Masked and permuted pre-training for language understanding, vol. 33. Curran Associates, Inc.; 2020, p. 16857–67.
[25] Nallapati R, Zhai F, Zhou B. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. 2016, arXiv:1611.04230.
[26] Zhang X, Lapata M, Wei F, Zhou M. Neural latent extractive document summarization. Brussels, Belgium: Association for Computational Linguistics; 2018, p. 779–84.
[27] Narayan S, Cohen SB, Lapata M. Ranking sentences for extractive summarization with reinforcement learning. 2018, arXiv:1802.08636.
[28] Srikanth A, Umasankar AS, Thanu S, Nirmala SJ. Extractive text summarization using dynamic clustering and co-reference on BERT. In: 2020 5th international conference on computing, communication and security. 2020, p. 1–5.
[29] Zhou Q, Yang N, Wei F, Huang S, Zhou M, Zhao T. Neural document summarization by jointly learning to score and select sentences. 2018, arXiv:1807.02305.
[30] See A, Liu PJ, Manning CD. Get to the point: Summarization with pointer-generator networks. Vancouver, Canada: Association for Computational Linguistics; 2017, p. 1073–83.
[31] Gehrmann S, Deng Y, Rush AM. Bottom-up abstractive summarization. 2018, arXiv:1808.10792.
[32] Celikyilmaz A, Bosselut A, He X, Choi Y. Deep communicating agents for abstractive summarization. New Orleans, Louisiana: Association for Computational Linguistics; 2018, p. 1662–75.
[33] Hsu W-T, Lin C-K, Lee M-Y, Min K, Tang J, Sun M. A unified model for extractive and abstractive summarization using inconsistency loss. 2018, arXiv:1805.06266.
[34] Liu Y, Lapata M. Text summarization with pretrained encoders. 2019, arXiv:1908.08345.
[35] Lin C-Y. ROUGE: A package for automatic evaluation of summaries. Barcelona, Spain: Association for Computational Linguistics; 2004, p. 74–81.
[36] Hermann KM, Kocisky T, Grefenstette E, Espeholt L, Kay W, Suleyman M, et al. Teaching machines to read and comprehend, vol. 28. Curran Associates, Inc.; 2015,

Junqing Fan received his Ph.D. degree from China University of Geosciences. He is currently an Associate Professor at the School of Computer Science, China University of Geosciences. He is a CCF member. His research interests include Data Mining, Natural Language Processing, High Performance Computing, and Optimization.

Xiaorong Tian received her B.S. degree in 2022 from China University of Geosciences, Wuhan, China, where she is currently working toward a master's degree in Electronic and Information Engineering. Her research interests include Deep Learning, Natural Language Processing, and Big data Technology.

Chengyao Lv is an Associate Professor in the School of Foreign Languages, China University of Geosciences. Her research interests are computational linguistics, Natural language processing, corpus-based research, and digital learning. She is the corresponding author of this article.

Simin Zhang holds a Bachelor's degree in Computer Science and Technology from Northwestern Polytechnical University and a Master's degree in Computer Technology from China University of Geosciences. His research interests include Deep Learning and Natural Language Processing.

Yuewei Wang received an M.S. degree in Computer Science and Technology in 2018 from China University of Geosciences, Wuhan, China, where he is currently working toward a doctoral degree in Geographic Information System. His research interests include Machine Learning, Big Data Analyze, Digital Earth, and High Performance Computing.

Junfeng Zhang Junfeng Zhang is Professor in the School of Foreign Languages, China University of Geosciences (Wuhan). His research interests include functional linguistics and translation studies.