J Inf Process Syst, Vol.16, No.4, pp.820~831, August 2020
https://doi.org/10.3745/JIPS.04.0181

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Joint Hierarchical Semantic Clipping and Sentence Extraction for Document Summarization

Wanying Yan* and Junjun Guo*

Abstract

Extractive document summarization aims to select a few sentences while preserving its main information on a given document, but the current extractive methods do not consider the sentence-information repeat problem especially for news document summarization. In view of the importance and redundancy of news text information, in this paper, we propose a neural extractive summarization approach with joint sentence semantic clipping and selection, which can effectively solve the problem of news text summary sentence repetition. Specifically, a hierarchical selective encoding network is constructed for both sentence-level and document-level document representations, and data containing important information is extracted on news text; a sentence extractor strategy is then adopted for joint scoring and redundant information clipping. This way, our model strikes a balance between important information extraction and redundant information filtering. Experimental results on both CNN/Daily Mail dataset and Court Public Opinion News dataset we built are presented to show the effectiveness of our proposed approach in terms of ROUGE metrics, especially for redundant information filtering.

Keywords

Extractive Summarization, Hierarchical Selective Encoding, Redundant Information Clipping

1. Introduction

News extractive summarization is one of the most important tasks in natural language processing (NLP), drawing much more attention recently especially in the public opinion domain. Document summarization tries to convert text or a collection thereof into a short summary containing key information. Generally, most of the traditional text summarization approaches focus only on summarizations without considering the sentence repeat problem. In news' summarization tasks, the main information has always been summarized very clearly; thus, sentence extraction strategy is the better choice.

At present, the main ways to complete the text summarization task are extractive summarization and abstract summarization. Abstract summarization generates summarization in a manner closer to manual work, which shows strong ability to represent, understand, and generate text. Note, however, that news' texts generally contain excellently constructed top sentences, usually done by extractive summarization methods [1]. For the prior art, the extractive type can better cover the original information. Extractive summarization involves extracting a few sentences with high similarity in the original text. The currently

Manuscript received March 20, 2020; first revision May 12, 2020; accepted May 24, 2020.

Corresponding Author: Junjun Guo (guojjgb@163.com)

* College of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China (yanwanying1996@163.com, guojjgb@163.com)

popular extractive summarization method involves scoring and selecting [2] or combining these two for learning [3] to complete the summary task by considering the similarity of the sentence and the meaning of the text. News text with repeated sentences could be observed according to the Chinese news text data used in our model, as shown in Table 1.

Table 1. Example of news text repeat sentence

Content	The Supreme People's Procuratorate issued a notice courtesy of **the two high and one department for fugitives to turn themselves in! Five situations can be considered automatic submission**. The Supreme Law, the Supreme People's Procuratorate, and the Ministry of Public Security jointly issued a "Notice Urging Fugitives to Surrender". The notice stipulates that those who flee from the date of publication of the notice until October 31, 2019 and who truthfully confess their crimes may be given lighter or mitigated punishment according to the law. **The two high and one department issue a notice for fugitives to turn themselves in! Five situations can be considered automatic submission** (section).
Summarization	The two high and one department issued a notice for fugitives to turn themselves in! Five situations can be considered automatic submission.

The bold text demonstrates a sentence with the same meaning in news text data.

For the repeated occurrence of the sentence described in the case, through random sampling and statistical analysis of 500 news texts, we can see that a considerable part of the news has more than two topic sentences. The specific statistical results are shown in Table 2.

Table 2. News text repeated sentence proportion

Topic sentence	Proportion (%)
1-S	48
2-S	26
N-S	5
0-S	21

As can be seen from Table 2, multi-topic sentences appeared repeatedly in news texts, accounting for more than 31%. Therefore, a redundant information clipping method should be used to complete the summary extraction to improve the performance of the summary results; if the currently popular extractive summary method is used, redundant summary sentences will be extracted, and both first and third sentences will be extracted; thus causing the summary information to be redundant as shown in Table 1. For massive news data, the extracted summarization contains too much redundant information, which will reduce the readability of the summarization. Therefore, to address news summarization problems, we propose a neural document summarization approach by joint learning to extract sentences and clip sentence semantic in a hierarchical manner. The main contributions of our work are to add bi-layer selective extraction based on important information filtering in the encoder and add a summary weight module of redundant information clipping to the sentence extractor.

Overall, our contributions can be summarized as follows:

- We propose a neural document summarization approach through joint hierarchical semantic clipping and sentence extraction.
- In order to solve this task, we first construct a hierarchical selective strategy in the encoder-subnetwork, with bi-layer selective extraction for reductant information filtering at both sentence and document levels.

- Sentence extraction is accomplished by joint sentence scoring and redundant information clipping at the document level.
- We also publish a Chinese news dataset on the public opinion of the courts. Experimental results show the effectiveness of our model.

2. Related Work

At present, automatic text summaries can be divided into extractive summaries and abstract summaries according to different implementation technologies [4]. The extractive method involves selecting keywords and key sentences from the original text by analyzing the statistical characteristics and subconscious semantic features of the text and generating a summary of the sentences or paragraphs. The abstractive method is based on understanding the semantics of the original text, condensing its main ideas to achieve semantic reconstruction, and subsequently constructing a new summary word-for-word [5]. Because news text contains artificially condensed topic sentences, using the extractive summary method can better complete the news text summary task. At present, extractive summary methods mainly include two categories: statistical-based methods and neural-based methods.

Statistical-based methods: Extractive summarization methods based on statistical learning are based mainly on the term frequency using various correlation calculations or using topic models and clustering-related tools. Extractive summarization methods by word frequency calculation mainly include LexRank, TextRank, LexPageRank, and GraphSum algorithms. For example, in the TextRank algorithm, preprocessed text sentences are regarded as nodes; an undirected weighted edge is constructed by similarity between sentences, iteratively updating the node value by the weights on the edge and finally selecting the N highest-scoring nodes as the final summary. The method of counting word frequency can use the global information on the entire document set, but its high complexity makes it unsuitable for the current task. In the clustering method, sentences in the article were encoded to obtain sentence-level vector representation and subsequently clustered to get different categories. Finally, the sentence closest to the centroid is used as the final summary. Extractive summarization methods based on statistical learning consider only the relationship between sentences, not the information of text semantics. Therefore, the emergence of neural network models effectively solves this problem.

Neural-based methods: Extractive summarization methods based on the neural network model are currently implemented mainly by sequence labeling and sentence ordering. Nallapati et al. [2] judge whether a sentence is a summarization sentence by labeling; Narayan et al. [6] do so based on the combination of Seq2Seq and reinforcement learning to complete the training of the model. Zhou et al. [3] proposed using sentence benefit as a scoring method and combined sentence scoring and sentence selection to complete the extractive summary task. Zhang et al. [7] used pre-trained Hierarchical Bidirectional Encoder Representations from Transformers to achieve sentence extraction.

In this paper, we propose a neural document summarization approach via joint hierarchical semantic clipping and sentence extraction to tackle the sentence redundant problem for long texts.

3. Document Summarization Model

Based on the importance of the news text information and data redundancy, we propose a neural

document summarization approach via joint hierarchical semantic clipping and sentence extraction (JhscSe). The JhscSe model is shown in Fig. 1.

It consists of a bi-layer selective encoder and a sentence extractor with redundant information filtering.

(1) Bi-layer selective encoder: It refers to the hierarchical selective encoder that includes sentence-level selective encoding and document-level selective encoding. It consists of a hierarchical encoder and a selective encoder. The document has a hierarchical mapping structure, i.e., words constitute a sentence, and sentences constitute a document. Most neural network models currently use hierarchical encoders to represent this hierarchical structure. For the text summary task, however, we want to select sentences containing the most important information. Thus, we added selective gate network mechanisms after sentence-level encoding, and document-level encoding constitutes a hierarchical selective encoder.

(2) Sentence extractor with redundant information filtering: It is composed of a redundant information clipping module and a sentence extractor. First, the encoded sentence is scored, and one of the sentences is extracted in each step. Second, the model remembers the output summary of the current steps when a summary is selected and updates the encoder information when scoring and selecting again. Finally, the information with the highest similarity to the summarization selected in the previous step is deleted, and the model is iterated until the limit of the number of output sentences is reached.

In this section, we will first introduce a bi-layer selective document encoder, followed by how to generate a summary through a sentence extractor filtered by redundant information.

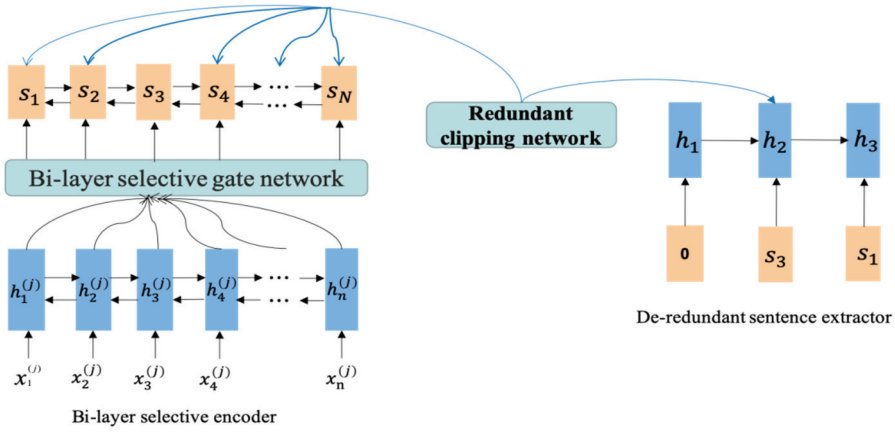

Fig. 1. JhscSe model.

3.1 Bi-layer Selective Encoder

Because the hierarchical model processes long texts as short texts, it can effectively solve the long-term dependencies of RNN (and its variants) and capture local and global semantic information. The data used in our experimental model is news text of different lengths. Therefore, we use hierarchical selective encoders of sentences and documents to represent the vector semantic representations of sentences and documents and enhance the display of important information in news text. Since a document $T = (s_1, s_2, \ldots s_N)$ contains N sentences, the sentence-level selective encoder first reads the j-th input sentence $S_j = (x_1^{(j)}, x_2^{(j)}, \ldots, x_n^{(j)})$ and constructs a basic sentence representation $\widetilde{s_j}$. The sentence-level selective encoder then selects a sentence containing important information words according to the semantic

information of sentence as input to the document encoder. Afterward, the document-level selective encoder selects documents containing important information sentences according to the semantic representation of document as shown in Fig. 2.

3.1.1 Sentence-level selective encoding processing

The sentence-level selective encoder first reads the input sentence and constructs the basic sentence representation. The sentence-level selective encoder by a BiGRU network reads the sentence word for word, and a BiGRU network consists of a forward GRU and a backward GRU. The forward GRU reads the word embedding of the words $(x_1, x_2, \dots x_n)$ in the sentence s_j from left to right and obtains a series of hidden states $(\vec{h}_1^{(j)}, \vec{h}_2^{(j)}, \dots, \vec{h}_n^{(j)})$; the backward GRU reads the word embedding input from right to left and gets another sequence of hidden states $(\overleftarrow{h}_1^{(j)}, \overleftarrow{h}_2^{(j)}, \dots, \overleftarrow{h}_n^{(j)})$:

$$\vec{h}_i^{(j)} = GRU(x_i^{(j)}, \vec{h}_{i-1}^{(j)}) \tag{1}$$

$$\overleftarrow{h}_i^{(j)} = GRU(x_i^{(j)}, \overleftarrow{h}_{i+1}^{(j)}) \tag{2}$$

The initial state of BiGRU is set to zero vector, $\vec{h}_i^{(j)} = \mathbf{0}$ and $\overleftarrow{h}_n^{(j)} = \mathbf{0}$.

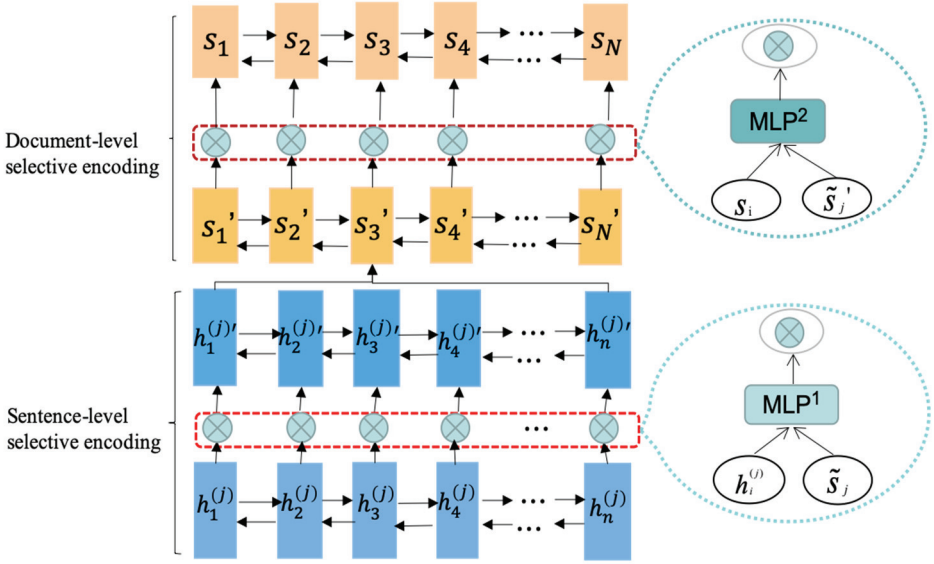

Fig. 2. Bi-layer selective encoder.

After reading the words $(x_1, x_2, \dots x_n)$ of the sentence S_j , we constructed its sentence-level representation $\widetilde{s_j}$ by concatenating the last forward and backward GRU hidden vectors:

$$\widetilde{s_j} = \begin{bmatrix} \overleftarrow{h}_1^{(j)} \\ \vec{h}_n^{(j)} \end{bmatrix} \tag{3}$$

To ease the burden of de-redundant sentence extractors and highlight the emergence of important information in text, we use the selective mechanism [8] in the encoder to complete the task of extractive summaries. Specifically, the selective gate network in our model uses two vectors as input: the sentence word vector $h_i^{(j)}$ and the sentence representation vector $\widetilde{s_j}$. As the output of the BiGRU encoder, the sentence word vector $h_i^{(j)}$ represents the meaning and context information of word x_i. The sentence vector $\widetilde{s_j}$ is used to represent the meaning of the sentence. For each word x_i, the selective gate network generates a selective gate vector $selectST_i$ using $h_i^{(j)}$ and $\widetilde{s_j}$, and then represents the hidden layer vector $h_i^{(j)\prime}$. For each time step *i*, the selective gate takes the sentence representation S_j and BiGRU hidden layer $h_i^{(j)}$as inputs to compute the selection gate vector $selectST_i$:

$$selectST_i = \sigma(W_s h_i^{(j)} + U_s \widetilde{s_j} + b) \tag{4}$$

$$h_i^{(j)'} = h_i^{(j)} \odot selectST_i \tag{5}$$

where W_s and U_s are weight matrices and b is the bias vector, with σ denoting a non-linear activation function. After the selective gate network, we obtain another sequence of vectors $(h_1^{(j)'}, h_2^{(j)'}, \dots, h_n^{(j)'})$. Selecting words containing important information, we construct a selectively encoded sentence-level representation $\widetilde{s_j}'$ by concatenating the last forward and backward GRU hidden vectors of the newly generated sequence, and then use this new sequence $\widetilde{s_j}'$ as the next document encoding input.

$$\widetilde{s_j}' = \begin{bmatrix} \overleftarrow{h_i}^{(j)\prime} \\ \overrightarrow{h_i}^{(j)\prime} \end{bmatrix} \tag{6}$$

3.1.2 Document-level selective encoding processing

We use another BiGRU as a document-level selective encoder to read sentences, read each article step by step, and get a representation of the document. Using a sentence-level $(\widetilde{s_1}, \widetilde{s_2}, \dots, \widetilde{s_n})$ encoded vector as input, the document-level encoder performs forward and backward GRU encoding and generates two lists of hidden layer vectors:$(\vec{s}_1, \vec{s}_2, \dots, \vec{s}_N)$ and$(\overleftarrow{s}_1, \overleftarrow{s}_2, \dots, \overleftarrow{s}_N)$. The document-level representation s_i of the sentence S_j is a concatenation of the forward and backward hidden layer vectors:

$$s_i = \begin{bmatrix} \vec{s}_N \\ \overleftarrow{s}_1 \end{bmatrix} \tag{7}$$

The main purpose of text summarization is to delete information. Therefore, we use another selective gate network in the document encoder to complete the task of extracting summarization. Specifically, the document-level selective gate network in our model takes two vectors as input: the sentence vector $\widetilde{s_j}'$ and the document representation vector s_i. As the output of the BiGRU sentence-level selective encoder, sentence vector $\widetilde{s_j}'$ represents the meaning and context information of sentence $\widetilde{s_j}$. The document representation vector s_i is used to represent the meaning of the document. For each sentence S_j, the document-level selective gate network generates a gate vector $selectDM_i$ with $\widetilde{s_j}'$ and s_i, and then regenerates the document representation vector s_i'.

For each time step i, the selective gate takes the output $\widetilde{s_j}'$ of the BiGRU sentence-level selective encoder and the document representation vector s_i as inputs to compute the document-level selection gate vector $selectDM_i$:

$$selectDM_i = \sigma(W_m\widetilde{s_j}' + U_m s_i + b) \tag{8}$$

$$s_i' = s_i \odot selectDM_i \tag{9}$$

where W_m and U_m are weight matrices and b is the bias vector, with σ denoting a non-linear activation function. After selecting the gate network at the document level, we obtain another sequence of vectors $(s_1', s_2', \ldots s_T')$ and subsequently use this new sequence as the input sentence representation of the decoder to complete the summary extraction.

$$s_T' = \begin{bmatrix} \vec{s_i}' \\ \overleftarrow{s_i}' \end{bmatrix} \tag{10}$$

Then, for the given document: $T = (s_1, s_2, \ldots s_N)$ obtains the final sentence vector.

3.2 Sentence Extractor with Redundant Information Filtering

The traditional process of the extractive summarization method involves labeling a sentence and then selecting it according to the relevant label or through scoring and selection by joint learning. Note, however, that the sentences extracted from the original text mostly contain repeat information. Therefore, the goal of our model is scoring and selection by jointly learning completely the summarization extracted as well as to reduce the occurrence of repeat information to improve the accuracy of the summarization. The two main functions of the decoder in this paper are as follows: (1) the next step of decoding memorizes the information of the sentence selected in the previous step; and (2) calculates its weight value according to the sentence $\tilde{s}_{N-1}$ selected in the first step and returns it to the encoder, updates the encoder information using its uncorrelated weight, and reduces the appearance of duplicate information, with the sentence extractor subsequently determining the next sentence $\tilde{s}_N$ by scoring the remaining document sentences. Our sentence extractor model, which is filtered based on redundant information, uses another GRU as a recursive unit to remember partial output summaries and utilizes a bi-layer perceptron (MLP) for scoring.

3.2.1 Joint scoring and selection considering redundant information filtering

As the specific process, the GRU takes the document-level representation s_{T-1}' of the sentence $\tilde{s}_{N-1}$ extracted by the last selective document encoding as input to generate its current hidden state h_t; the sentence scorer is a bi-layer MLP that uses two input vectors: the current hidden state h_t and the sentence representation vector s_{T-1}'.The score $\varphi(S_T)$ of the sentence s_T' is calculated by a non-linear activation function.

$$h_t = GRU(s_{T-1}', h_{t-1}) \tag{11}$$

$$\varphi(S_T) = W_n \tanh(W_q h_t + W_s s_T') \tag{12}$$

Among them, W_n, W_q , W_s are learnable parameters. When extracting the first sentence, we use another non-linear activation function to initialize the GRU hidden state h_0.

$$h_0 = \tanh\left(W_f \overleftarrow{s_1'} + b_f\right) \tag{13}$$

$$S_T = \emptyset \tag{14}$$

$$s_0 = 0 \tag{15}$$

where W_f, b_f is the learnable parameter and $\overleftarrow{s_1'}$ is the last backward state of the selective document-level encoder BiGRU. We use a zero vector to represent the previously extracted sentences s_0, since no sentences have been extracted.

Redundant clipping modules: Since most of the sentences extracted from the original text contain repetitive information, we use the non-linear activation function to calculate the weight ∂ of the current sentence by concatenating the currently selected sentence s_{T-1} with all the information $s_T{}'$ of the original text when the first sentence is selected. The irrelevant information $(1-\partial)$ is used to update the encoder information, and the current hidden state h_t and the sentence representation vectors $s_{T-1}{}'$ are used to calculate the score $\varphi(S_T)'$ of the sentence after de-redundant information through a non-linear activation function.

$$\partial = sigmoid[s_{T-1}, s_{T-1}{}'] \tag{16}$$

$$s_T{}' = (1-\partial) * s_T{}' \tag{17}$$

$$\varphi(S_T)' = W_n \tanh\left(W_q h_t + W_s s_T{}'\right) \tag{18}$$

For the score of all sentences at time *s*, we choose the sentence with the largest gain score.

$$\hat{S}_s = argmax(\varphi(S_T)') \ , \ S_T \in T \tag{19}$$

3.2.2 De-redundant objective function

The activation function was applied to our de-redundant model to improve the performance of the model. We first use the normalized exponential function softmax to normalize the predicted sentence scores $\varphi(S_T)'$ so that model prediction distribution P can be obtained. Then, we use another softmax function to calculate and generate labeled data distribution D as our training target.

In deep learning, Kullback–Leibler (KL) divergence is used to evaluate the difference between the predicted value distribution and the true value distribution of the model output. Therefore, we use KL as our loss function and minimize KL loss function l by relative entropy:

$$l = D_{kl} = (P \| D) \tag{20}$$

4. Experiment

4.1 Data

Two datasets are used in our model. The first set of data is about TNTICO (News Text Involving Court

Opinion), obtained by crawling from major websites (such as WeChat, Weibo, etc.) and through our data cleaning in a certain way to get it. The second set of data used a CNN/Daily Mail dataset (proposed by Hermann et al. [9]). The TNTICO dataset is court-oriented public opinion news information characterized by the topic sentence getting summarized very accurately and repeating itself, which can be considered a task in a specific field. The CNN/Daily Mail dataset is mainly derived from articles and related issues of the news network and daily news suitable for common domain tasks. We performed labeled data preprocessing on two sets of data, using the same preprocessing data method as that used by Zhou et al. [3]. The relevant information of the two datasets is shown in Tables 3 and 4.

Table 3. TNTICO dataset

	Doc_n	DocSentence_n	DocLength	TgtSenTence_n	TgtLength
train	78k	9.26	647.48	1.28	28.32
dev	2.6k	8.74	667.19	1.62	26.83

Table 4. CNN/Daily Mail dataset

	Doc_n	DocSentence_n	DocLength	TgtSenTence_n	TgtLength
train	0.287M	31.58	791.36	3.79	55.17
dev	13.37k	26.72	769.26	4.11	61.43

4.2 Experimental Parameter Settings

We use the Glove tool to pre-train word vectors; the dimension of the word vector is 50 dimensions, and the TNTICO dataset and CNN/Daily Mail dataset vocabularies are set to 30,000 and 60,000, respectively. The word embedding size, BiGRU-based double-layer selective encoding layer, and de-redundant information sentence extractor are set to 50, 256, and 256 dimensions, respectively. We use a Gaussian distribution with the Xavier scheme to initialize the parameters randomly and utilize Adam as the optimization algorithm, with the learning rate finally set to 0.001 through the continuous parameter adjustment of the experiment and the dropout set to 0.2. In training and testing, we set each article to 80 sentences and set the length of each sentence to 100 words. The model in this article is implemented by PyTorch.

4.3 Baseline

We first compared with the following five baseline models and performed ablation experiments on our model to show the effectiveness of each module of our model:

(1) **Lead3**: Generally, the first part of the article is more important, containing the overall information of the article. The Lead3 method involves selecting the first three sentences of the text as the final summary.
(2) **TextRank**: Summarization is done by selecting sentences of higher importance in the text.
(3) **Nmf**: This method is non-negative supervised matrix factorization. It constructs a weight matrix using the relationship between sentences and topics and sorts the topic probabilities of each sentence to complete the summarization extracted.
(4) **SummaRunner**: This model is a recurrent neural network model based on a sequence classifier. This extraction model is trained using a generative training mechanism.

(5) **NeuSum**: This model uses the encode–decode method to combine sentence ordering and selection in an end-to-end model to complete the summarization extracted.

4.4 Experimental Results

4.4.1 Evaluation metrics

This article uses ROUGE at evaluation metrics. It is now widely used in text summary evaluation tasks. The criterion of the ROUGE method is to evaluate the summarization based on the co-occurrence information of n-grams in the abstract. The basic idea is to use the n-tuple contribution statistics of the summarization generated by the model and the reference summarization as evaluation metrics, with the quality of the summarization evaluated by counting the number of overlapped basic units (n-grams, word sequences, and word pairs) of these two. In our model, we use the "f" values of Rouge-1, Rouge-2, and Rouge-L as evaluation metrics.

4.4.2 Analysis of results

(1) Comparative experiment: The JhscSe model and five baselines in this paper make up different comparison experiments on the CNN/Daily Mail dataset and the TNTICO dataset as shown in Tables 5 and 6.

According to the results in Tables 5 and 6, the three experimental models (Lead3, TextRank, and Nmf) are traditional machine learning methods; the three other models, SummaRunner, NeuSum, and JhscSe, are neural network methods. It can be seen that the neural network-based model has an average improvement of 7.5 percentage points over the machine-based learning method according to the rouge score. Therefore, the neural network-based extractive summarization method is more effective than the traditional machine learning method. Compared with SummaRunner, NeuSum, and JhscSe, the Rouge score is improved by about 1–2 percentage points; thus, a neural document summarization approach via joint hierarchical semantic clipping and sentence extraction can better improve the model's summary performance.

Table 5. TNTICO dataset experiments

Model	Rouge-1	Rouge-2	Rouge-L
Lead3	36.37	27.10	24.57
TextRank	35.71	26.39	23.62
Nmf	42.13	32.19	34.31
SummaRunner	43.22	33.45	32.67
NeuSum	45.61	36.87	35.14
JhscSe	46.98	37.29	35.25

Table 6. CNN/Daily Mail dataset experiments

Model	Rouge-1	Rouge-2	Rouge-L
Lead3	39.20	15.70	35.50
TextRank	40.20	17.56	36.44
Nmf	39.68	16.21	36.13
SummaRunner	39.60	16.20	35.30
NeuSum	40.11	17.52	36.69
JhscSe	40.85	17.61	36.72

(2) Ablation study: The JhscSe model in this paper has made different improvements in the encoder and decoder. The joint scoring and selection based on the bi-layer selective encoding according to importance screening and redundant information reduction mechanism are improved. The experimental results are shown in Table 7.

According to the experimental results, a hierarchical selective strategy in the encoder-sub network (with bi-layer selective extraction for reductant information filtering at both sentence and document levels) and the sentence extraction accomplished by joint sentence scoring and redundant information clipping at the document level can effectively improve the performance of the model and solve the problem of summarization sentence redundancy.

Table 7. Ablation experiment of the TNTICO dataset

Model	Rouge-1	Rouge-2	Rouge-L
JhscSe-select	45.74	36.89	35.16
JhscSe-(1-a)	46.47	36.95	35.11
JhscSe	46.98	37.29	35.25

5. Conclusion and Future Work

In this study, we have proposed JhscSe, a new abstractive summarization method for the long text summarization problem that aims to extract summaries on joint learning of importance selection and redundant information clipping. We first added hierarchical selective encoding strategies to the encoder; this method enhances the appearance of important information in the text, and then adds a de-redundancy mechanism to the sentence extractor to reduce the text containing repeat information. The experimental results on both Chinese and English datasets show that our proposed method effectively solves the repeat information problem and has a significant improvement compared to the baseline models. Nonetheless, our proposed method can only be used for single document summarization. In the next step, we will apply the redundancy module to multi-document data.

Acknowledgement

This study was supported by the project of the National Key Research and Development Program of China (No. 2018YFC0830100), National Natural Science Foundation of China (No. 61762056, 61866020), Natural Science Foundation Project of the Yunnan Science and Technology Department (No. 2019FB082), and Personal Training Project of the Yunnan Science and Technology Department (No. KKSY2017 03015).

References

[1] J. Cheng and M. Lapata, “Neural summarization by extracting sentences and words,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany, 2016.

[2] R. Nallapati, F. Zhai, and B. Zhou, "SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, San Francisco, CA, 2017.

[3] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "Neural document summarization by jointly learning to score and select sentences," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, Melbourne, Australia, 2018, pp. 654-663.

[4] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review*, vol. 47, no. 1, pp. 1-66, 2017.

[5] Z. Cao, W. Li, S. Li, and F. Wei, "Retrieve, rerank and rewrite: soft template based neural summarization," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, 2018, pp. 152-161.

[6] S. Narayan, S. B. Cohen, and M. Lapata, "Ranking sentences for extractive summarization with reinforcement learning," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, New Orleans, LO, 2018, pp. 1747-1759.

[7] Q. Zhou, N. Yang, F. Wei, and M. Zhou, "Selective encoding for abstractive sentence summarization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, Vancouver, Canada, 2017, pp. 1094-1104.

[8] X. Zhang, F. Wei, and M. Zhou, "HIBERT: document level pre-training of hierarchical bidirectional transformers for document summarization," in *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019)*, Florence, Italy, 2019, pp. 5059-5069.

[9] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," *Advances in Neural Information Processing Systems*, vol. 28, pp. 1693-1701, 2015.

Wanying Yan **https://orcid.org/0000-0002-8244-8954**

She received B.S. degree in the School of Biomedical Engineering of South-Central University for Nationalities in 2018 and is pursuing a master's degree at the School of Information Engineering and Automation of Kunming University of Science and Technology since September 2018. Her current research interests include natural language processing and text summaries.

Junjun Guo **https://orcid.org/0000-0002-3522-7120**

In July 2010, he graduated from China University of Petroleum (East China) with a B.S. degree in engineering in automation; in June 2017, he graduated from Xi'an Jiaotong University with Ph.D. degree in engineering in control science and engineering. He is currently a lecturer in the Department of Automation, Kunming University of Science and Technology, Yunnan. His research interests include multi-source information fusion, natural language processing, multi-modal machine translation, etc.