

# A Hybrid Method of Extractive Text Summarization Based on Deep Learning and Graph Ranking Algorithms

SHI Hui<sup>1</sup>, WANG Tiexin<sup>1,2\*</sup>

1. College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, P.R. China;

2. Key Laboratory of Safety Critical Software, Ministry of Industry and Information Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, P.R.China

(Received 8 May 2022; revised 23 July 2022; accepted 28 August 2022)

**Abstract:** In the era of Big Data, we are faced with an inevitable and challenging problem of “overload information”. To alleviate this problem, it is important to use effective automatic text summarization techniques to obtain the key information quickly and efficiently from the huge amount of text. In this paper, we propose a hybrid method of extractive text summarization based on deep learning and graph ranking algorithms (ETSDG). In this method, a pre-trained deep learning model is designed to yield useful sentence embeddings. Given the association between sentences in raw documents, a traditional LexRank algorithm with fine-tuning is adopted in ETSDG. In order to improve the performance of the extractive text summarization method, we further integrate the traditional LexRank algorithm with deep learning. Testing results on the data set DUC2004 show that ETSDG has better performance in ROUGE metrics compared with certain benchmark methods.

**Key words:** extractive text summarization; deep learning; sentence embeddings; LexRank

**CLC number:** TP 391      **Document code:** A      **Article ID:** 1005-1120(2022)S-0158-08

## 0 Introduction

With the rapid development of Internet in the 21st century and the exponential growth of text data, it has become an urgent problem for users to quickly and effectively distill the required useful information from the massive information<sup>[1]</sup>. To alleviate this problem, text summarization technologies can be used to help users quickly grasp the content of the original text. As a fundamental and important task in natural language processing (NLP), the main goal of text summarization is to use computers to automatically distill a short, concise and coherent paragraph from a long text or collection of texts that reflects the central content of the source text.

There are two main forms of text summarization generation, namely extractive and abstractive. Abstractive summarizations require complex lan-

guage processing and generate sentences with poor readability. In contrast, extractive summarizations generate sentences with better readability and now more maturely developed. Hence, in this paper, we focus on extractive summarizations. Extractive summarizations, as the name implies, consist of abstracting key text units, including but not limited to words, phrases and sentences, etc., from original texts. Extractive text summarizations usually retain the salient information of the source text, have correct syntax, and are suitable for long text summarizations.

Most existing extractive summarization methods use sentences as the basic text unit for extraction, since they are the smallest grammatical unit that can be expressed. However, this kind of method usually faces two challenges: (1) How to extract and represent sentences correctly? (2) How to

\*Corresponding author, E-mail address: tiexin.wang@nuaa.edu.cn.

**How to cite this article:** SHI Hui, WANG Tiexin. A hybrid method of extractive text summarization based on deep learning and graph ranking algorithms[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2022, 39(S):158-165.

<http://dx.doi.org/10.16356/j.1005-1120.2022.S.021>

select and sort sentences that reflect the meaning of the source text?

To address these two problems, we propose a new extractive text summarization method: A hybrid method of extractive text summarization combining deep learning and graph ranking algorithms (ETSDG). ETSDG is a combination of a deep learning pre-trained model and the traditional algorithm LexRank. The main contributions of this work are as follows:

(1) To better express sentences, we propose a new sentence representation method using deep learning technologies. A pooling operation is added to yield interpretable sentence embeddings.

(2) To select the sentences that are more rele-

vant to the whole text, we obtain a threshold of 0.3 for setting the best result while fine-tuning in ETS-DG.

(3) We propose a hybrid method which integrates the traditional graph ranking algorithm with deep learning technologies.

## 1 Related Work

Current mainstream extractive text summarization techniques can be classified into unsupervised learning methods (feature scoring, graph ranking, etc.) and supervised learning methods (classification algorithms, deep learning, etc.). Table 1 shows a brief conclusion of related works based on whether they are supervised or not.

**Table 1** Brief conclusion of related works

Technology	Note	Limitation	Type
Feature scoring	Select key words and phrases to form a summarization based on word frequency, sentence position, or similarity to the first sentence	Need to set weights manually, low quality	Unsupervised learning
Graph ranking	Based on sentence similarity	Relatively slow operation	Unsupervised learning
Classification algorithms	Use SVM, Bayesian and other classification models to determine whether a sentence belongs to a summarization	Ignoring the connection between sentences	Supervised learning
Deep learning	Sentence extraction using neural network models such as CNN and RNN	Higher requirements for data, easy to have the effect of gradient disappearance, etc	Supervised learning

Unsupervised learning methods aim to generate summarizations by ranking sentences. Most of early unsupervised methods extracted summarizations by analyzing features of original texts, which include word frequency, similarity of the first sentence to the title, and elements such as sentence length and sentence centrality. The work of Luhn<sup>[2]</sup> used word frequency features to solve the text summarization task, which aimed mainly to find those sentences containing the most information to compose the summarization. This approach is simple and fast, but the effect is susceptible to anomalous data generating summarizations that are not related to the topic.

With the continuous development of extractive summarization research, a variety of unsupervised extractive summarization methods have emerged. Typical unsupervised learning methods are graph ranking methods. Graph ranking-based methods,

such as TextRank<sup>[3]</sup>, rely on sentence similarity<sup>[4]</sup>. Mihalcea et al.<sup>[3]</sup> introduced the extraction of sentences with high importance in the text to form text summarizations by TextRank. This method is concise and effective, but it can lead to a slower operation because it is subject to arbitrary sentence similarity calculations and iterative calculations.

Supervised learning methods treat text summarization as a classification problem. Traditional machine learning algorithms, which belong to supervised methods, are applied to extractive text summarization. Extractive summarizations can be converted into a traditional natural language processing problem of text classification. Classification-based methods use classification models such as support vector machine (SVM)<sup>[5]</sup>, Bayesian<sup>[6]</sup> to determine whether sentences are affiliated with summarizations. Louis<sup>[7]</sup> combined background knowledge on

the basis of Bayesian Surprise model to form summarizations. The effectiveness of this method depends on the quality of training data and domains of applications, etc.

In recent years supervised extractive summarization methods are more based on neural networks, where neural network models can learn vector representations of words and sentences to form abstract features, and then use these features for ranking. Yin et al.<sup>[8]</sup> firstly trained a representation of sentences using a convolutional neural network (CNN) language model, and then used the PageRank algorithm to calculate sentences importance and iteratively selected the important sentences. Cao et al.<sup>[9]</sup> synthesized the features obtained by using recurrent neural networks (RNN) learning and traditional features for sentence ranking. Abdelsalam et al.<sup>[10]</sup> conducted a study on the performance of variants of BERT-based models on text summarization and proposed “SqueezeBERTSum”. These deep learning-based methods can learn text unit features better, but they have high data requirements, have more parameters, and are prone to gradient disappearance and other effects<sup>[11-12]</sup>.

ETSDG is inspired from the work of Yin et al.<sup>[8]</sup> Representations of sentences are first trained using an advanced pre-trained model, and then the LexRank algorithm is used to calculate sentence importance and select more representative sentences to form a summarization. Experiment results show our method can be used for text summarizations effectively.

## 2 Main Work

The framework of the proposed ETSDG is shown in Fig. 1. The main modules of the framework are linked with solid lines. The input and output artifacts in modules are linked with dotted lines. The goal of ETSDG is to extract a short, concise and coherent paragraph from a long text or collection of texts that reflects the central content of the source text. Specifically, ETSDG includes five main modules (1)—(5) shown in Fig. 1.

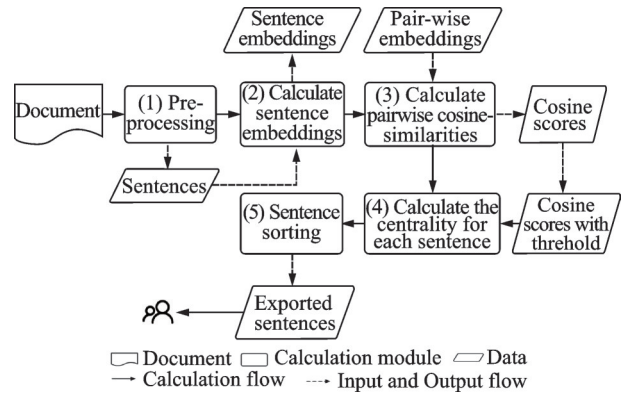


Fig.1 Framework of the proposed ETSDG

- (1) Pre-process raw documents to get sentences;
- (2) Calculate sentence embeddings using deep learning;
- (3) Calculate the pair-wise cosine similarities;
- (4) Calculate the centrality for each sentence;
- (5) Sort sentences so that the first selected sentence will be the one with the highest score.

### 2.1 Document pre-processing

In the first step, we pre-process raw documents. ETSDG uses sentences as the basic text unit for extraction and splits the raw document into sentences with the separator “.”, “?” and “!”, etc. However, sentence splitting is not a simple problem. The punctuation marks “?” and “!” can split sentences without having ambiguous meanings. But “.” has different uses and is not necessarily the end of a sentence, so we need to exclude some situations. For example, “Mr. and Mrs. Smith” went to the ball. In this sentence, “Mr. And Mrs. Smith” is a person phrase, where the “.” is part of the person phrase, then it is not the end of the sentence. To solve this punctuation ambiguity, a simple rule of text break is used in ETSDG, namely “.” is not preceded by an abbreviation and is not followed by a number, then it indicates that the end of the sentence is detected.

### 2.2 Calculation of sentence embeddings

There are dozens of proposed neural network-based methods to calculate sentence embeddings, such as Skip-Thought<sup>[13]</sup>, InferSent<sup>[14]</sup> and Universal Sentence Encoder<sup>[15]</sup>, etc. These methods are usually trained starting from a random initializa-

tion<sup>[16]</sup>. They come at the cost of a dramatic increase in calculation time and memory usage with sentence length.

In order to reduce calculation time, memory usage and better express sentences, we propose a new model based on a pre-trained DistilRoBERTa network. A pooling operation is added in ETSDG to yield interpretable sentence embeddings.

ETSDG first uses a DistilRoBERTa network, which is a pre-trained Transformer network, and then fine-tune it to get sentence embeddings. The basic model Transformer replaces recurrent layers with self-attention in an encoder-decoder framework and has achieved state-of-the-art results in language modeling. Based on Transformer, ETSDG aims to eliminate recurrence to allow more direct interaction among words in a sentence.

Moreover, a pooling operation is added to the output of DistilRoBERTa to derive sentence embeddings, which helps ETSDG reduce calculation time and memory usage.

Based on this pre-trained deep learning model, ETSDG can map sentences or paragraphs to a 768 dimensional dense vector space such that similarity sentences are close. Fig.2 shows the architecture of calculating sentence embeddings.

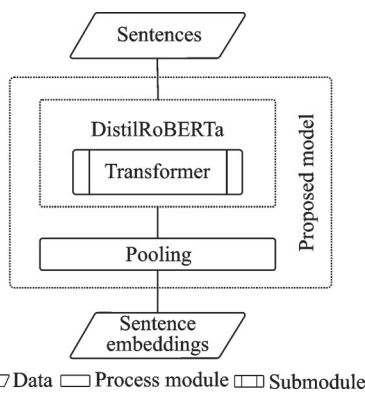


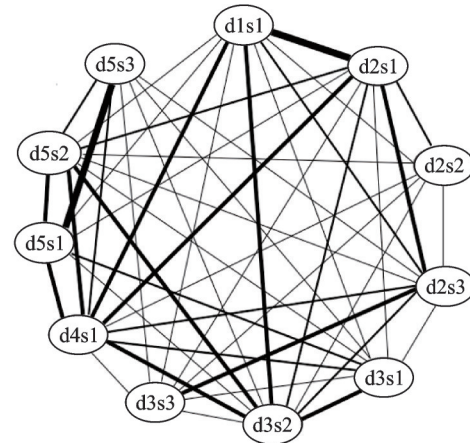
Fig.2 Architecture of calculating sentence embedding

### 2.3 Calculation of pair-wise cosine similarities

After transforming sentences into sentence embeddings, we aim to consider similarities between sentences more effectively and exclude the influence of noisy sentences that have little to do with the center of the source text. Thus, we calculate similarities between pairwise sentences based on LexRank<sup>[17]</sup>.

LexRank algorithm is a graph theory-based natural language processing method proposed by Erkan et al., which focuses on generating summarizations by filtering out summarization sentences through the judgment of similarity between sentences.

As shown in Fig.3, the LexRank algorithm processes the sentences and constructs a scalar graph. In Fig. 3, nodes represent sentences and an edge between two nodes represents the similarity of two sentences. If two sentences are unrelated, there is no edge between the corresponding nodes. If two sentences have a greater similarity, edge between this pair of sentence is given a larger value. We call values of edges as edge weights.



Edge weights: **—** [0.3,1.0] **—** [0.2,0.3] **—** [0.1,0.2] **—** [0.0,0.1]

Fig.3 Schematic diagram of LexRank algorithm<sup>[17]</sup>

ETSDG uses cosine similarity to calculate the similarity of pair-wise sentences. The score of the pair-wise cosine similarity ranges from 0 to 1. According to scores of the pair-wise cosine similarities, a connection matrix between sentences is formed to iteratively calculate the amount of information contained in sentences.

### 2.4 Calculation of centrality for each sentence

The main goal of this calculation module is to find key sentences of documents by calculating the centrality for each sentence. When calculating the sentence centrality, we fully consider the number of connecting edges of nodes corresponding to each sentence and weights of the connecting edges, i.e., the size of the centrality and relevance of the sentence. Finally, according to a certain threshold, sentences with a higher score are selected as the key

sentences of the document.

In order to better set the threshold in ETSDG, we conduct an experiment on a benchmark dataset “DUC2004 Task2”. In the process of fine-tuning, a pilot study is experimented to set the threshold ranging from 0 to 1.0. We set the final threshold according to the scores of commonly used text summarization automatic evaluation indicators ROUGE-1, ROUGE-2, and ROUGE-L. The experiment results are shown in Fig. 4. Due to the decreasing trend of experimental results after 0.5, Fig. 4 only shows the results with adjustment thresholds between 0 and 0.5.

As shown in Fig. 4, when we set the threshold as “0.3”, scores of ROUGE-1, ROUGE-2 and ROUGE-L are better. Hence, we finally set the threshold of “0.3” in ETSDG. With this “0.3” as threshold, ETSDG selects sentences as key sentences of documents.

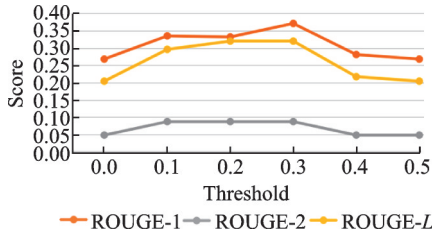


Fig.4 Testing results of ETSDG with different thresholds

## 2.5 Sentence sorting

In the last calculation module, ETSDG sorts sentences considering the centrality of each sentence and edge weights in the text graph structure to generate summarizations. More representative sentences are positioned higher in the final generated summarization. The Equation we used in sentence sorting is described as follows

$$\text{weight}(u) = \frac{d}{N} + (1-d) \times \sum_{v \in \text{adj}[u]} \frac{\text{weight}(v)}{\text{deg}(v)} \quad (1)$$

where  $\text{weight}(u)$  represents the centrality of sentence  $u$ , and  $d$  the “damping factor”, which is usually set to 0.85.  $N$  represents the total number of sentences,  $\text{adj}[u]$  the set of sentences that are adjacent to  $u$ , and  $\text{deg}(v)$  the degree of the sentence  $v$ .

Based on Eq. (1), we finally select salient sentences to form a summarization.

## 3 Experiments

In this section, we introduce eight benchmark methods, the data set “DUC”, the evaluation method “ROUGE”, experiment results and analyses of the proposed ETSDG.

### 3.1 Set up

In order to evaluate the performance of ETSDG, we select eight commonly used benchmark methods (i. e., Lead Baseline, Random, GraphRank, FreqSum, TsSum<sup>[18-22]</sup>) and the models that performed better on the DUC2004 task (BasicSum and TSSM<sup>[23]</sup>) for comparison.

Since some of these eight methods use different metrics in their evaluation, in order to be able to reasonably compare with them, this paper adopts the results on the DUC2004 dataset (from Ying et al.<sup>[23]</sup>), and directly quotes the results of these benchmark models to evaluate each method using uniform evaluation method ROUGE<sup>[24]</sup>.

### 3.2 Data set

Document understanding conference (DUC) is a commonly used dataset for measuring text summarization methods and an important evaluation criterion applicable to measuring extractive text summarization tasks. Therefore, in this paper, the DUC dataset is chosen as the main corpus for experiments.

Considering that DUC-2004 is the most commonly used comparison dataset in the DUC corpus, we conducted experiments with DUC-2004 to evaluate the performance of ETSDG. It contains 500 documents, each with four different manually generated reference summarizations.

### 3.3 Evaluation method

Current evaluation methods are divided into automatic evaluation and manual evaluation methods according to whether there are manual participations involved. A widely metric used in automatic evaluation methods is ROUGE. Hence, in this paper, we choose ROUGE to evaluate the performance of ETSDG.

ROUGE is an automatic digest evaluation

method proposed by Lin<sup>[24]</sup>. The basic idea of ROUGE is to compare the systematic summarizations generated by the model with the reference summarizations. Commonly used evaluation metrics are ROUGE-1, ROUGE-2, and ROUGE- $L$ , etc., where 1, 2, and  $L$  stand for 1-word, 2-word and longest substring based, respectively. This method is one of the common criteria of summarization evaluation systems, and its calculation formula is

$$R_{\text{ROUGE-N}} = \frac{\sum_{S \in \{\text{Ref}\}} \sum_{N_{n\text{-gram}} \in S} \text{Count}_{\text{match}}(N_{n\text{-gram}})}{\sum_{S \in \{\text{Ref}\}} \sum_{N_{n\text{-gram}} \in S} \text{Count}(N_{n\text{-gram}})} \quad (2)$$

where  $n$ -gram denotes the  $n$ -word,  $\{\text{Ref}\}$  the reference summarization, and  $\text{Count}_{\text{match}}(N_{n\text{-gram}})$  the number of  $n$ -gram appearing in both the system summarization and the reference summarization, and

$\text{Count}(N_{n\text{-gram}})$  the number of  $n$ -gram appearing in the reference summarization.

In the evaluation phase, we used the toolkit pyrouge to calculate the ROUGE scores of ETSDG, using commonly used evaluation metrics ROUGE-1, ROUGE-2, ROUGE- $L$  for comparison with other benchmark methods.

### 3.4 Results and discussion

We compare ETSDG on DUC2004 using ROUGE evaluation method with eight benchmark methods. Table 2 lists results<sup>[19,23]</sup> of these methods along with ETSDG. The values of ROUGE indicate the number of overlapping text units between generated summarizations and reference summarizations. More overlapping text units represent better performance of methods.

**Table 2 Performance comparison on DUC2004 using ROUGE evaluation method**<sup>[19,23]</sup>

Parameter	Lead Baseline	Random	GraphRank	Centroid	FreqSum	TsSum	BasicSum	TSSM	ETSDG
ROUGE-1	0.292	0.290	0.304	0.364	0.353	0.359	0.366	0.369	<b>0.372</b>
ROUGE-2	0.043	0.041	0.041	0.075	0.081	0.815	0.091	0.089	<b>0.088</b>
ROUGE- $L$	0.271	0.264	0.265						<b>0.321</b>

Data from the second to the fourth column are extracted from Ref.[19], Data from the fifth to the ninth column are extracted from Ref.[23].

As shown in Table 2, the overall performance of ETSDG is better than the other eight methods. From the results, it can be inferred that better performance can be obtained with text summarization techniques using a hybrid method.

Lead Baseline is simple and outperforms Random in all of ROUGE measures, however, it only selects the leading sentences of a document to generate a summarization while ignoring connections between sentences. GraphRank takes connections between sentences into consideration and performs a little better than Lead Baseline in terms of ROUGE-1 metric. However, since GraphRank fails to better express sentences, it gets poor performance in terms of ROUGE-2 and ROUGE- $L$  metrics. The above three traditional methods, Lead Baseline, Random and GraphRank, perform much worse than our proposed ETSDG due to their single feature considered.

Although the data of Centroid, FreqSum and TsSum methods are significantly better than the

three traditional methods mentioned above for ROUGE-1 and ROUGE-2, they still inferior to ETSDG. The reason is that these three approaches fail to output sentences embeddings well and thus the overall performance is poor, while ETSDG does a better job in representing sentence embeddings.

BasicSum and TSSM slightly outperform ETSDG on ROUGE-2, but still underperform ETSDG on ROUGE-1. The reason for the gap in ETSDG on ROUGE-2 may lie in the fact that it only considers correlations between sentences, without further considering common words frequency features and named entities in sentences. These are also some of the directions we will consider to further optimize ETSDG in the future.

Compared with eight benchmark methods above, the overall performance of our proposed ETSDG is still good. It not only better expresses sentences using a pretrained model, but also considers connections between sentences using LexRank.

Moreover, the threshold of “0.3” we set in ETSDG helps us both filter out some weakly similar sentences and avoid losing too many sentences related to the source text. These reasons explain why ETSDG can obtain better performances.

Experiment results show our method which is a fusion of deep learning and traditional LexRank algorithms can be used for text summarizations effectively.

## 4 Conclusions

The social value of automatic text summarization, which belongs to the domain of text generation in natural language processing, demonstrates the importance of automatic text summarization in the natural language domain. In this paper, a hybrid approach combining deep learning and graph ranking algorithms is introduced from the perspective of extractive methods. ETSDG firstly uses a pre-trained language model to represent sentence embeddings, secondly fully considers connections between sentences in documents based on the LexRank algorithm with a threshold, and finally finds more representative sentences to express the source text. According to the general text summarization evaluation method ROUGE, the results on “DUC2004” data set show that ETSDG has a better performance than certain benchmark methods. In the future, we hope to generate better summarizations using richer semantic information and leveraging external knowledge.

## References

- [1] LI Jinpeng, ZHANG Chuang, CHEN Xiaojun, et al. Survey on automatic text summarization[J]. Journal of Computer Research and Development, 2021, 58(1): 1-21.
- [2] LUHN H P. The automatic creation of literature abstracts[J]. IBM Journal of Research and Development, 1958, 2(2): 159-165.
- [3] MIHALCEA R, TARAU P. Textrank: Bringing order into text[C]//Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. Stroudsburg: ACL, 2004: 404-411.
- [4] PAGE L, BRIN S, MOTWANI R, et al. The PageRank citation ranking: Bringing order to the web: SIDL-WP-1999-01-20 [R].[S.l.]: Stanford InfoLab, 1999.
- [5] CORTES C, VAPNIK V. Support vector machine[J]. Machine learning, 1995, 20(3): 273-297.
- [6] ITTI L, BALDI P F. Bayesian surprise attracts human attention[J]. Advances in Neural Information Processing Systems, 2005, 18: 547-554.
- [7] LOUIS A P. A bayesian method to incorporate background knowledge during automatic text summarization[C]//Proceedings of the 52nd Annual Meeting of the ACL. Stroudsburg: ACL, 2014: 333-338.
- [8] YIN W, PEI Y. Optimizing sentence modeling and selection for document summarization[C]// Proceedings of Int Joint Conference on Artificial Intelligence. Menlo Park: AAAI, 2015: 1383-1389.
- [9] CAO Z, LI W, LI S, et al. Improving multi-document summarization via text classification[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park: AAAI, 2017: 3053-3059.
- [10] ABDEL-SALAM S, RAFEA A. Performance study on extractive text summarization using bert models[J]. Information, 2022, 13(2): 67.
- [11] WANG Ershen, SONG Yuanshang, XU Song, et al. ADS-B anomaly data detection model based on deep learning and difference of Gaussian approach[J]. Transactions of Nanjing University of Aeronautics & Astronautics, 2020, 37(4): 550-561.
- [12] YADAV A K, SINGH A, Dhiman M, et al. Extractive text summarization using deep learning approach[J]. International Journal of Information Technology, 2022, 14(5): 1-9.
- [13] KIROS R, ZHU Y, SALAKHUTDINOV R R, et al. Skip-thought vectors[J]. Advances in Neural Information Processing Systems, 2015, 28: 3294-3302.
- [14] CONNEAU A, KIELA D, SCHWENK H, et al. Supervised learning of universal sentence representations from natural language inference data[C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics, 2017: 670-680.
- [15] CER D, YANG Y, KONG S, et al. Universal sentence encoder for English[C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Brussels, Belgium: Association for Computational Linguistics, 2018: 169-174.
- [16] REIMERS N, GUREVYCH I. Sentence-BERT: Sentence embeddings using siamese BERT-Net

- works[J]. arxiv Preprint arxiv: 1908.10084, 2019.
- [17] ERKAN G, RADEV D R. Lexrank: Graph-based lexical centrality as salience in text summarization[J]. Journal of Artificial Intelligence Research, 2004, 22: 457-479.
- [18] DING Xiaofei. Research on the generation of review summary based on text summarization technology [D]. Changsha: Hunan University, 2019.
- [19] HAN X, LV T, HU Z, et al. Text summarization using FrameNet-based semantic graph model[J]. Scientific Programming, 2016. DOI: 10.1155/2016/5130603.
- [20] HAN X, LV T, JIANG Q, et al. Text summarization using sentence-level semantic graph model[C]// Proceedings of 2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS). [S.l.]: IEEE, 2016: 171-176.
- [21] NENKOVA A, VANDERWENDE L, MCKEOWN K. A compositional context sensitive multi-document summarizer: Exploring the factors that influence summarization [C]// Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. [S.l.]: ACM, 2006: 573-580.
- [22] LIN C T, HOVY E. The automated acquisition of topic signatures for text summarization[J]. Proceedings of the 18th Conference on Computational Linguistics. Saarbrücken, Germany: Coling, 2000: 495-501.
- [23] YIN Wenhao, LI Sujian, SUI Zhifang. A topic-sensitive extractive method for multi-document summarization [J]. Journal of Chinese Information Processing, 2017, 31(6): 155-161.
- [24] LIN C Y. Rouge: A package for automatic evaluation of summarizations[C]// Proceedings of the Workshop of ACL 2004. Stroudsburg: ACL, 2004: 74-81.

**Authors** Ms. SHI Hui is a master student in Nanjing University of Aeronautics and Astronautics, with research interests in semantic relevance computing and text summarization. Dr. WANG Tiexin has research interests in knowledge engineering, semantic computing, etc. He is a member of CCF (69740M).

**Author contributions** Ms. SHI Hui designed the study, compiled the models and wrote the manuscript. Dr. WANG Tiexin reviewed the draft. All authors commented on the manuscript draft and approved the submission.

**Competing interests** The authors declare no competing interests.

(Production Editor: LIU Yandong)

## 结合深度学习和图排序算法的抽取式文摘方法

史 荟<sup>1</sup>, 王铁鑫<sup>1,2</sup>

(1. 南京航空航天大学计算机科学与技术学院, 南京 211106, 中国;

2. 南京航空航天大学工业和信息化部安全关键软件重点实验室, 南京 211106, 中国)

**摘要:** 大数据时代面临着一个不可避免的挑战性问题“信息过载”。为了缓解这一问题, 必须使用有效的自动文本总结技术, 从海量的文本中快速有效地获取关键信息。本文提出了一种结合深度学习和图排序算法的抽取式文本总结 (Extractive text summarization based on deep learning and graph ranking algorithms, ETSDG) 的混合方法。在这种方法中, 一个预先训练好的深度学习模型被设计用来产生有用的句子嵌入。此外, 考虑到原始文档中句子之间的关联, 在 ETSDG 中采用了传统的 LexRank 算法并进行了微调。为了提高抽取式文本总结方法的性能, 进一步将传统的 LexRank 算法与深度学习相结合。在数据集 DUC2004 上的测试结果表明, 与某些基准方法相比, ETSDG 在 ROUGE 指标上有更好的表现。

**关键词:** 抽取式文本总结; 深度学习; 句子嵌入; LexRank