Keyword Acquisition for Language Composition Based on TextRank Automatic Summarization Approach

Yan Jiang*, Chunlin Xiang, Lingtong Li
Department of Primary Education, Chongqing Preschool Education College, Chongqing, 404047, China

***Abstract*—It is important to extract keywords from text quickly and accurately for composition analysis, but the accuracy of traditional keyword acquisition models is not high. Therefore, in this study, the Best Match 25 algorithm was first used to preprocess the compositions and evaluate the similarity between sentences. Then, TextRank was used to extract the abstract, construct segmentation and named entity model, and finally verify the research content. The results show that in the performance test, the Best Match 25 similarity algorithm has higher accuracy, recall rate and F1 value, the average running time is only 2182ms, and has the largest receiver working characteristic curve area, which is significantly higher than other models, reaching 0.954. The accuracy of TextRank algorithm is above 90%, the average accuracy of 100 text analysis is 94.23%, the average recall rate and F1 value are 96.67% and 95.85%, respectively. In comparison of the application of the four methods, the research model shows obvious advantages, the average keyword coverage rate is 94.54%, the average processing time of 16 texts is 11.29 seconds, and the average 24-hour memory usage is only 15.67%, which is lower than the other three methods. The experimental results confirm the superiority of the model in terms of keyword extraction accuracy. This research not only provides a new technical tool for language composition teaching and evaluation, but also provides a new idea and method for keyword extraction research in the field of natural language processing.**

Keywords—Language composition; keywords; best match 25; textrank; digests

I. Introduction

In today's digital era, natural language processing (NLP) technology plays an increasingly important role in the field of text analysis. Keyword extraction, as a basic text analysis tool, is important for understanding and processing large amounts of text data [1]. Especially in the field of education, efficient and accurate keyword extraction has great application value for the analysis and evaluation of language composition (LC). Traditional keyword extraction methods, such as term frequency-inverse document frequency (TF-IDF), latent delicacy allocation (LDA), graph-based lexical rank (LexRank) algorithms, have been applied in several fields, but still have limitations in efficiency and accuracy in specific scenarios [2-3]. TextRank's automatic summarization (AS) method extracts key sentences from text in an unsupervised learning manner in a concise and efficient way, which can be achieved without a large amount of labeled data. This algorithm has good adaptability and scalability, can be applied to texts in different domains, and is easily integrated with other NLP techniques [4]. As a graph-based algorithm, TextRank also reveals the text structure, increasing the depth of analysis and content level understanding. In view of this, research has focused on exploring LC keyword extraction techniques based on the TextRank AS method [5]. The goal of the study is to improve the efficiency and accuracy of LC keyword extraction by optimizing and applying the TextRank algorithm. The significance of the study is as follows: first, TextRank-based AS not only improves the efficiency of the keyword extraction process and reduces the workload of teachers, but also enhances the objectivity and consistency of the evaluation. Second, the study not only expands the application scope of the algorithm in Chinese text processing, but also promotes the innovative application of language processing technology in the field of education, and provides a new perspective for practical problem solving of NLP technology in language education. In addition, with the wide application of artificial intelligence in various industries, AI-assisted language teaching and assessment is becoming an emerging trend. By optimizing the keyword extraction process, this study lays the foundation for building a smarter educational assistance system, which further promotes the development of AI in the field of educational technology. In summary, this study has far-reaching research significance in enhancing teaching efficiency, promoting technological innovation, and leading the development of educational technology. The study is divided into four main parts, the first of which is a detailed description of relevant studies in recent years. In the second part, the main methods of the experiment are firstly introduced. The third part is to verify the validity and reliability of the research model through experimental design and data analysis. The fourth part is to summarize and prospect the research.

In the study of AS for text, K. E. Dewi and N. I. Widiastuti developed an AS model for Indonesian text that aims to reduce the number of sentences while retaining key information. The model utilized three summarization methods: extractive, abstractive, and hybrid. Extractive selected key sentences, abstract reconstructed new sentences to describe the content, while hybrid combines the advantages of both. The system design consisted of a pre-processing (sentence segmentation, tokenization, co-reference parsing, deactivation, feature extraction) and a processing phase (selecting and arranging important sentences and words to form a summary). The model was particularly suitable for document input and adaptation to long text and multi-document input is a direction for further research [6]. H. Aliakbarpour et al. proposed a new model of abstract summarization combining convolutional neural networks with long and short-term memory and incorporating the auxiliary attention mechanism of an encoder to enhance the saliency and fluency of the summaries. Tested on CNNDaily Mail and DUC-2004 datasets, the model outperformed the benchmark model in terms of ROUGE score, saliency and readability [7]. Y. Huang et al. proposed a novel elemental graph augmented abstract summarization model for the challenges of legal opinion journalism AS. The model utilized pre-trained language model reinforcement sequences and structural encoders to extract key information through a network of structural graphs and graph transformers to effectively guide the decoding process. Tests on a legal opinion news corpus revealed that the model outperforms other baselines in terms of ROUGE and BERT scores, and its effectiveness was proven by manual evaluation [8]. A. Zagar and M. Robnik-Sikonja presented a cross-language AS approach to summarizing Slovenian news articles using a pre-trained English summary model. To address decoder limitations, additional language models were introduced for target language text evaluation. The cross-language model was demonstrated to be qualitatively similar to the target language-specific model through automatic and manual evaluation, but occasionally misleading or absurd content appeared [9]. E. Inan proposed an entity-based text summarization method that recognizes named entities and constructs dependency graphs from a pre-trained language model. A reconciliation centrality algorithm was applied to summarize the entity ordering, outperforming the unsupervised learning baseline and approaching the state-of-the-art end-to-end model [10].

In summary, the recent literature in the field of automatic text processing, especially in keyword extraction and summary generation, has demonstrated several notable advances. Researchers have developed different approaches in order to accommodate multiple languages and text formats. For example, Dewi and Widiastuti developed a model containing multiple summarization techniques specifically for Indonesian text to accommodate long texts and complex documents. In the widely studied TextRank algorithm, Qiu and Zheng enhanced its performance in keyword extraction through tolerance rough sets, while Hernawan et al. improved the accuracy of the algorithm in sentence importance assessment using BM25. Huang and Xie improved the accuracy of keyword extraction for patented text by combining the TextRank algorithm with a priori knowledge networks. Further, Aliakbarpour et al. combined a convolutional neural network and a long and short-term memory network while incorporating an attention mechanism to enhance the quality of text summarization. The elemental graph augmented abstract summarization model proposed by Huang et al. on the other hand, demonstrates superiority in handling legal opinion news. Given the potential application of TextRank in automatic keyword extraction, the study proposes to use this algorithm to extract keywords for LC. It is expected to further improve the algorithm's ability and accuracy in extracting key contents for Chinese essays by improving TextRank. This will not only provide support for automatic scoring of compositions, but also help educators to have a more comprehensive understanding of students' writing skills and content focus, so as to provide more effective guidance and feedback.

II. Language keyword Acquisition model Based on TextRank Algorithm

The study, in order to construct a language keyword acquisition model with higher accuracy, first preprocesses the LC by BM25 similarity algorithm, and the query calculates the similarity between the LC sentences. Then the automatic digest results of the composition corpus are obtained based on TextRank algorithm. After that, the construction of participle model and named entity model is carried out. Finally, the description related to dictionary design and keyword acquisition strategy is unfolded.

A. *Abstract Acquisition Based on BM25 Similarity Algorithm with Textrank Algorithm*

In the study of keyword acquisition, Best Match 25 (BM25) similarity algorithm is an algorithm used in information retrieval and text mining to measure the relevance between a query and a document [11]. Its advantages include effectiveness against long documents, ability to handle documents of different lengths, automatic adjustment of the weights of query terms, ability to handle scarce terms, and efficiency in large text collections [12]. The BM25 algorithm has been widely used in the field of information retrieval, and is able to more accurately assess the relevance between documents and queries, and therefore has significant practical value in large-scale document retrieval and search engines [13]. The general formula of BM25 similarity algorithm is shown in Eq. (1).

$$Score(Q,d)=\sum_{i}^{n}W_i.R(q_i,d) \tag{1}$$

In Eq. (1), Q denotes the sentence to be retrieved, q_i denotes the morpheme obtained from the sentence, and W_i denotes the weight of q_i. d denotes the target sentence, and R denotes the relevance score of q_i and d. When q_i occurs more times in the sentence, it means that the similarity weight it represents decreases, and in order to avoid the result error caused by this situation, there exists an expression as shown in Eq. (2).

$$IDF(q_i)=\log\frac{N-n(q_i)+0.5}{n(q_i)+0.5} \tag{2}$$

In Eq. (2), N denotes the total number of sentences and $n(q_i)$ denotes the number of q_i sentences included. The formula for similarity is shown in Eq. (3).

$$R(q_i,d)=\frac{f_i.(k_1+1)}{f_i+K}.\frac{qf_i.(k_2+1)}{qf_i+k_2} \tag{3}$$

In Eq. (3), k_1 and k_2 denote the conditioning factor constants, and qf_i denotes the number of times the word appears in the retrieval process [14]. If the number of times the word appears in the process of retrieval is 1, the formula can be further simplified, as shown in Eq. (4) [15].

$$R(q_i, d) = \frac{f_i.(k_1+1)}{f_i + K} \tag{4}$$

In Eq. (4), f_i denotes the frequency of occurrence of words, and the expression of K is shown in Eq. (5).

$$K = k_1.(1-b+b.\frac{dl}{avgdl}) \tag{5}$$

In Eq. (5), dl denotes the length of the sentence, then $avgdl$ denotes the average length of all sentences, and b is a constant and represents the moderating factor [16-17]. Querying the correlation between texts through the BM25 similarity algorithm mitigates the interference encountered in calculating the similarity. Therefore, the study used the BM25 similarity algorithm to preprocess the LC, after which the keywords and abstracts were obtained by Textrank algorithm. TextRank algorithm is a graph-based text summarization method that determines keywords and sentences in text by analyzing the interconnections between words in the text. The algorithm first represents the text as a node graph, then calculates the weights of the nodes through the connection relationship between the nodes, and uses iterative computation to gradually update the weights of the nodes, and finally determines the keywords and sentences. The advantages of the TextRank algorithm include a structured representation of the text, the ability to capture semantic associations between words, applicability to multilingual texts, no restriction on the length of the text, the ability to handle unsupervised learning, and it has proven its effectiveness and usefulness in the field of text summarization and keyword acquisition [18]. Therefore, the TextRank algorithm has important application prospects in automatic text summarization and keyword acquisition tasks. The operation principle of Textrank algorithm is shown in Fig. 1.

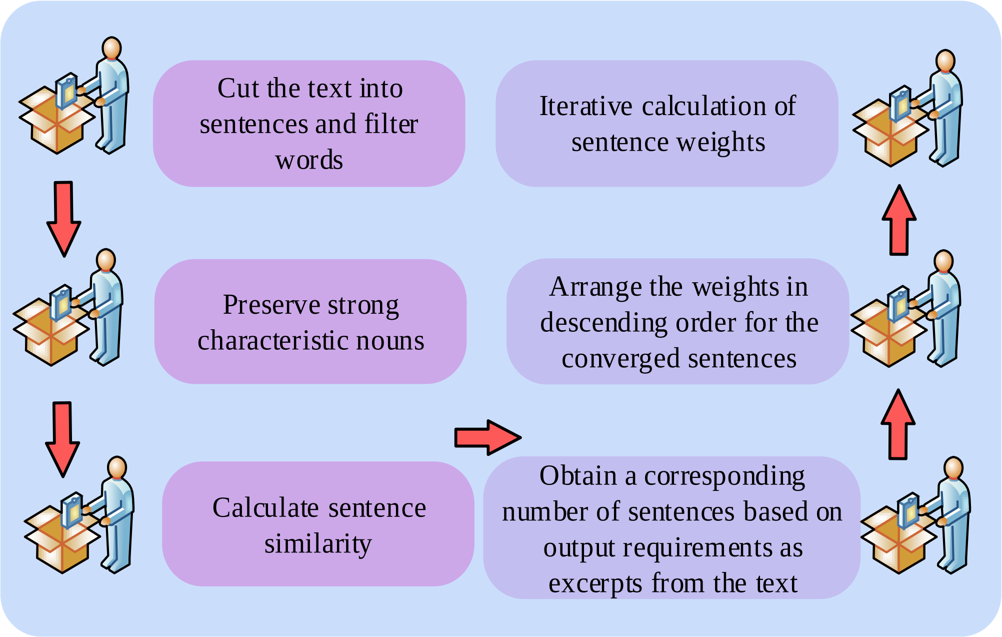

Cut the text into sentences and filter words
Preserve strong characteristic nouns
Calculate sentence similarity
Obtain a corresponding number of sentences based on output requirements as excerpts from the text
Arrange the weights in descending order for the converged sentences
Iterative calculation of sentence weights

Fig. 1. Operating principle of textrank algorithm.

In Fig. 1, firstly, the text is segmented into sentences and partitioned, while filtering out the stop words and retaining the characteristic words such as nouns and adjectives. Secondly, the similarity between sentences is calculated, the graph structure is constructed, the sentences are taken as nodes on the graph, and the value of the edges is the similarity between the sentences. Then, the weight of each sentence node is determined by iterative calculation. Finally, the descending order is sorted according to the weights, and the sentence with the highest weight is selected as the digest sentence. Textrank algorithm is divided into two parts: calculating similarity and sorting, in the calculation of similarity based on the idea of PangRank to build a graph, based on the nodes of the graph to calculate the similarity between sentences as shown in Eq. (6).

$$Score(V_i) = (1-d_1) + d_1 * \sum (W_{ji} / O_j) \tag{6}$$

In Eq. (6), d_1 denotes the damping coefficient, which usually takes the value of 0.85, meaning the probability of going from one graph node to another, with the purpose of avoiding the node weight values in the fringes from being assigned to 0. W_{ji} denotes the weight of node V_i pointing to node V_j, and O_j denotes the out-degree (i.e., number of edges connected out) of node V_j. $\sum$ denotes the cumulative summation operation for all nodes V_j pointing to node V_i C [19].

B. *Construction of the Disambiguation Model, Named Entity Model*

After the summarization process, the composition information has removed most of the redundant data, and then the keywords can be obtained from it. Firstly, the coverage of keywords should be set, and the model of word splitting is constructed from the keyword acquisition and elaborated on the basis of named entity model and self-built lexicon. After that, the strategy of LC keyword acquisition is proposed. Segmentation model is a model used in NLP to segment a continuous text sequence into meaningful units. In Chinese text processing, the role of the participle model is particularly significant, because there is no obvious word separator symbol like space in Chinese. Therefore, word segmentation modeling applies various techniques and methods, such as rule-based segmentation, lexicon-based segmentation, and statistical and machine learning-based segmentation [20]. The research adopts dictionary-based participle modeling, and the N-shortest path participle algorithm is one of the popular algorithms. N-Shortest Path Segmentation Algorithm is a Chinese segmentation algorithm based on graph theory and dynamic programming, compared with the traditional shortest path segmentation algorithm, it can better deal with problems such as ambiguity and unregistered words, and its characteristics are more suitable for discriminative named entity recognition, the operation principle of N-Shortest Path Algorithm is shown in the schematic diagram in Fig. 2.

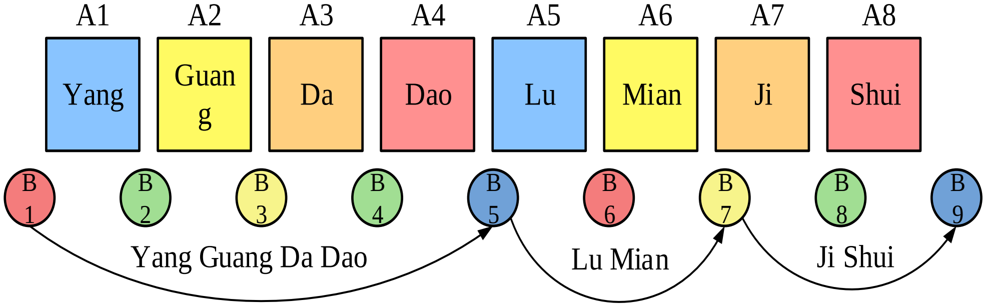


Fig. 2. Schematic diagram of the operating principle of the N-shortest path algorithm.

Fig. 2 shows an example subsection of the N-shortest path algorithm in operation, where A denotes each Chinese character present in the sentence and B denotes the node in the graph. This sentence demonstrates the ideal path: "Yang guang da dao/Lu mian/Ji shui" after the step of disambiguation, i.e., B1 to B5, B5 to B7, and B7 to B9 are recognized as reasonable paths. Meanwhile, words such as "Dao lu" and "Mian ji" also represent a path, and the algorithm first finds the shortest N paths of the sentence, and then calculates the most probable result based on the shortest paths. If noise interference is encountered, the word cut sign is lost, resulting in a situation where the output is a Chinese character string A. The existence of the formula is shown in Eq. (7).

$$P(W \mid C) = P(W) * \frac{P(C \mid W)}{P(C)} \tag{7}$$

In Eq. (7), W denotes the result sought after the improvement, and P denotes the probability that the partition result is divided correctly. $P(W \mid C)$ denotes the probability that the word string becomes a string of Chinese characters, and the value of $P(C)$ is kept constant. On the basis of maintaining the independence between sentences and introducing the unitary processing model in the n-gram model, the existence formula is shown in Eq. (8).

$$P(W) = \prod_{i=1}^{m} p(W_i) \tag{8}$$

In Eq. (8), it is assumed that each word occurs with equal probability and all are $p(W_i)$, where m denotes the number of words in the sentence. To summarize, the operation of the N-shortest path algorithm is mainly divided into three steps: in the first step, the sentence to be split into words is constructed into a directed graph, in which each node represents a word or words, and the edges between the nodes indicate the transfer relationship between words or words, with corresponding weights on each edge. In the second step, all possible paths in the graph are traversed and the weights of the paths are calculated using dynamic programming. The optimal n paths are found by recording the predecessor nodes of each node and maintaining a priority queue of path lengths. In the third step, based on the obtained optimal paths, path merging is performed to obtain the final disambiguation result. After the construction of the participle model, a cascading hidden Markov model (HMM) based named entity recognition is proposed for the processed corpus. HMM is a probabilistic model for modeling time-series data and is commonly used in speech recognition, NLP and bioinformatics [21-22]. It consists of a hidden Markov chain and a sequence of observations, where the hidden Markov chain represents the sequence of states of the system and the sequence of observations represents the sequence of observations dependent on each state [23]. The principle of operation of the HMM model with the labeling of person and place name roles is shown in Fig. 3.

Fig. 3(a) illustrates the operational steps of the HMM model, where the set of hidden states, the set of observations, the initial state probability distribution, and the state transfer probability distribution are first determined and used to generate the sequence of hidden states. Then the probability distributions of the observations are generated based on the hidden states to generate the corresponding observation sequences [24]. After that, the model parameters, including the initial state probability distribution, the state transfer probability distribution, and the observation generation probability distribution, are learned from the known observation sequences. Finally, with the given model and observation sequences, the Viterbi algorithm or forward-backward algorithm is utilized to decode or predict the most probable hidden state sequences [25-26]. Fig. 3(b) shows the role labeling of the model for the case of role labeling of person and place names. In the HMM model, for the case that the words do not appear in the existing lexicon, the method of calculating the output probability based on the role-word generation model is proposed, whose expression is shown in Eq. (9).

$$P(w|c)=\prod_{j=0}^{k} p(w_{p+j}|r_{p+j})p(r_{p+j}|r_{p+j-1}) \tag{9}$$

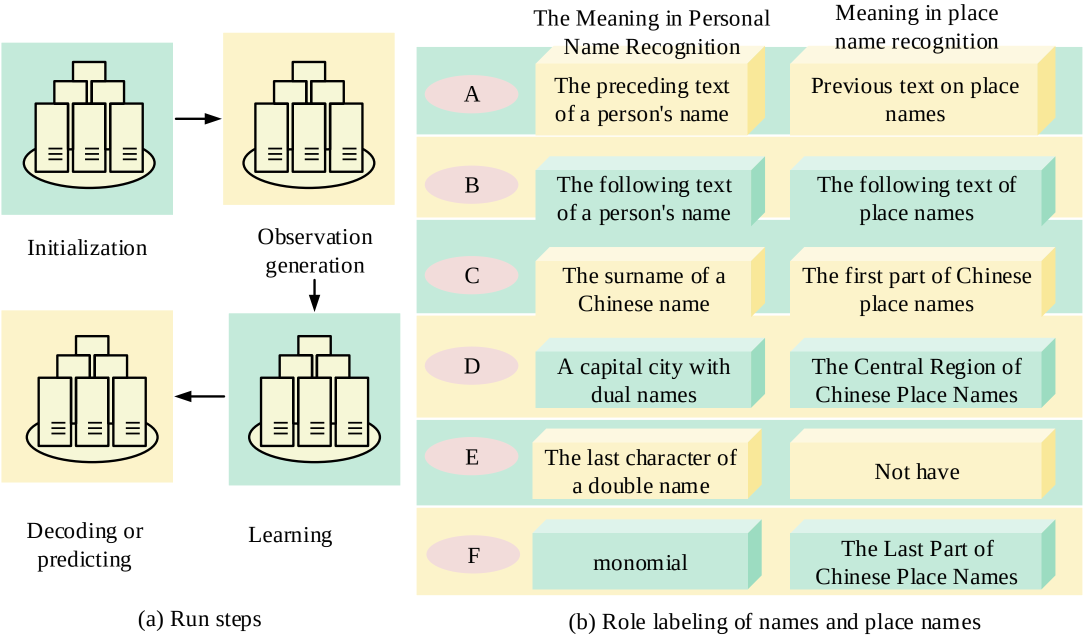

Initialization
Observation generation
Decoding or predicting
Learning
(a) Run steps
The Meaning in Personal Name Recognition
Meaning in place name recognition
A
The preceding text of a person's name
Previous text on place names
B
The following text of a person's name
The following text of place names
C
The surname of a Chinese name
The first part of Chinese place names
D
A capital city with dual names
The Central Region of Chinese Place Names
E
The last character of a double name
Not have
F
monomial
The Last Part of Chinese Place Names
(b) Role labeling of names and place names

Fig. 3. Operating principle of HMM model and labeling of person and place name roles.

In Eq. (9), w denotes a word that is not in the dictionary, r denotes a collection of roles. c denotes the category of the entity, $p(r_{p+j}|r_{p+j-1})$ denotes the transfer probability between the previous and previous roles, and $p(w_{p+j}|r_{p+j})$ denotes the probability that w occurs in r. The HMM model can be used to identify the important names and places in the LC corpus. The HMM model is able to effectively identify important names of people and places in the LC corpus, which can be used to assist in the acquisition of keywords for compositions [27-28].

C. Keyword Acquisition Strategy

The study uses the open source natural language framework HanLP to implement the use of entity recognition and segmentation, in order to improve the accuracy of keyword acquisition, the study uses a customized dictionary, the self-built dictionary for characters and scenery is shown in Fig. 4.

In Fig. 4, in the recognition of characters, they are categorized with the help of different types of nouns. And in the recognition of scenery, it is defined with the help of affixes. After that, the language corpus is analyzed to obtain the keywords, which have a restricted vocabulary of five, and classified for their connotations: article type, core, and key description." Article type" usually refers to the genre of the essay, such as argumentative essay, narrative essay, expository essay, application essay, etc., each of which has different writing characteristics and structures and is used to express different purposes and emotions." Core" usually refers to the theme or center of the essay, which is the main idea or argument that the writer wants to express, and it represents the focus and core of the essay [29-30]. "Key description" refers to the part of the composition that describes the core entities in detail, which may include the description of things, the characterization of characters, the narration of events, and the development of the relevant plot, etc. These descriptions usually occupy an important place in the composition to highlight the central idea and content of the essay. These descriptions usually occupy an important position in the composition to highlight the central idea and content of the essay. The process of keyword acquisition is shown in Fig. 5.

In Fig. 5, the recognition of named entities is performed based on the coarsely-scored segmentation results, which result in words being labeled lexically. Then the comprehensive deactivation word list adopted for eliminating deactivated words aims to eliminate words that are commonly used in LC and to reduce the interference with keyword acquisition. This is followed by entity statistical analysis and finally keyword acquisition. Among them, the analysis process of word lexicality is shown in Fig. 6.

In the presentation in Fig. 6, a two-stage process for core entity acquisition can be seen. First, the system uses standard named entity recognition techniques to identify entities. Next, in the case that the lexical label of an entity is a person's name or a place's name, the system checks whether the counter of the corresponding category has reached the upper limit of two entities. If the category to which a word belongs already has two entities, the word will not be processed further. If the upper limit has not been reached, the word is added to the final result set. This process ensures that entities are effectively identified and categorized, while limiting the number of entities in each category, keeping the result set streamlined and relevant. If an analyzed term is not included in the self-constructed lexicon, it will be included in the evaluation of the key descriptions, and the description rules are shown in Fig. 7.

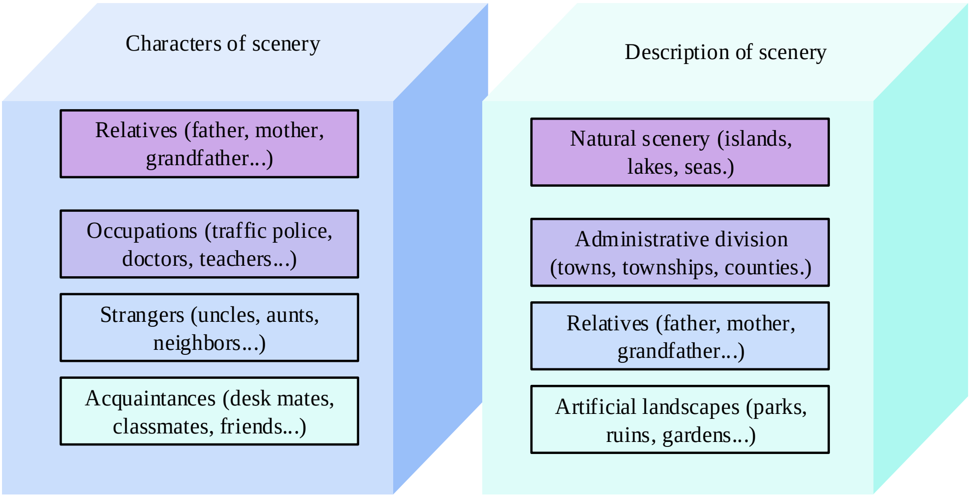

Characters of scenery
Relatives (father, mother, grandfather...)
Occupations (traffic police, doctors, teachers...)
Strangers (uncles, aunts, neighbors...)
Acquaintances (desk mates, classmates, friends...)
Description of scenery
Natural scenery (islands, lakes, seas.)
Administrative division (towns, townships, counties.)
Relatives (father, mother, grandfather...)
Artificial landscapes (parks, ruins, gardens...)

Fig. 4. Self-built dictionary of characters and scenery.

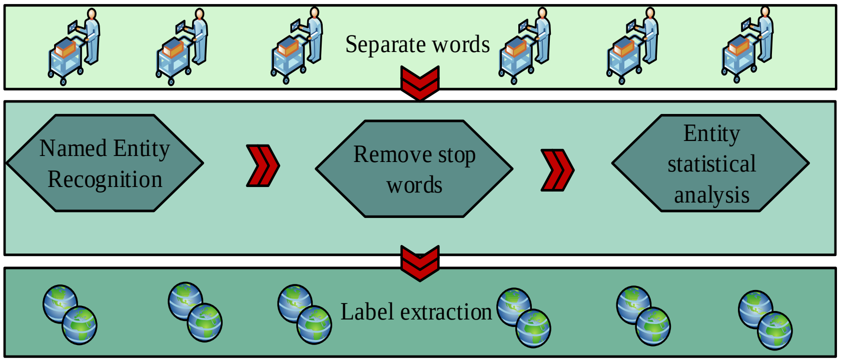


Fig. 5. Keyword acquisition process.

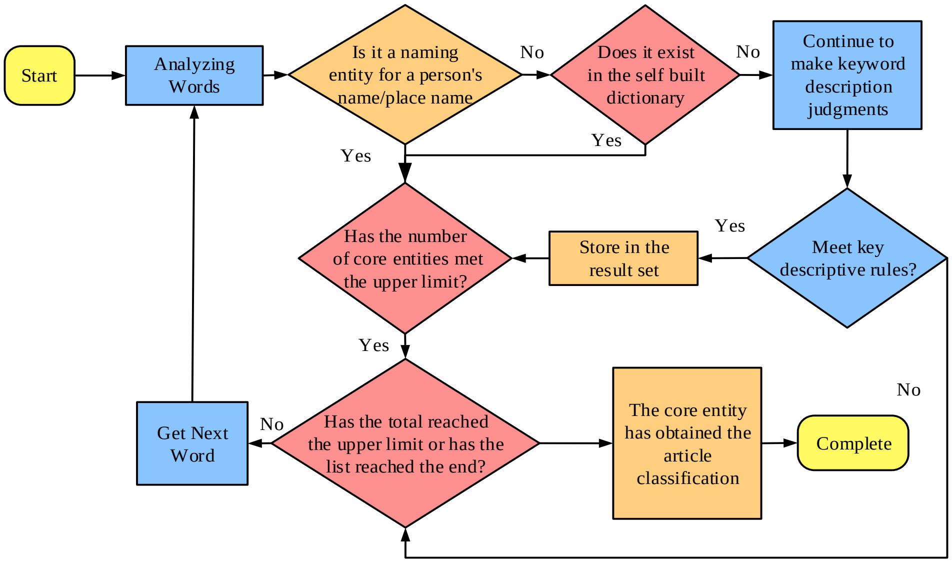
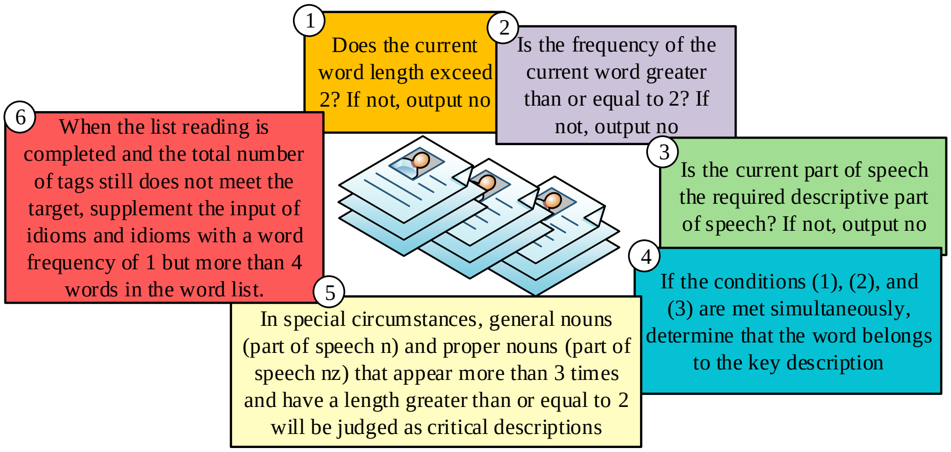

Start
Analyzing Words
Is it a naming entity for a person's name/place name
No
Does it exist in the self built dictionary
No
Continue to make keyword description judgments
Yes
Yes
Has the number of core entities met the upper limit?
Store in the result set
Yes
Meet key descriptive rules?
Yes
No
Get Next Word
No
Has the total reached the upper limit or has the list reached the end?
The core entity has obtained the article classification
Complete

Fig. 6. Analysis process of word parts of speech.

Based on the rules in Fig. 7, the key description section is obtained, and finally, the system performs a comprehensive analysis of the word list. Once the number of keywords extracted from the list meets the requirements, or when the end of the word-phrase list is read, the word-list analysis is completed. In addition, the type keywords for the articles were determined by comparing the weights of the two main types of named entities, person names and place names. If names were given more weight than places, the article was categorized as a "characterization". If names of places were given more weight, the article was categorized as "description of scenery". In the case of equal weight, both keywords will be added to the result set. Through this series of steps, the keyword acquisition of elementary school essays can be successfully completed.



1 Does the current word length exceed 2? If not, output no

2 Is the frequency of the current word greater than or equal to 2? If not, output no

3 Is the current part of speech the required descriptive part of speech? If not, output no

4 If the conditions (1), (2), and (3) are met simultaneously, determine that the word belongs to the key description

5 In special circumstances, general nouns (part of speech n) and proper nouns (part of speech nz) that appear more than 3 times and have a length greater than or equal to 2 will be judged as critical descriptions

6 When the list reading is completed and the total number of tags still does not meet the target, supplement the input of idioms and idioms with a word frequency of 1 but more than 4 words in the word list.

Fig. 7. Key description rules.

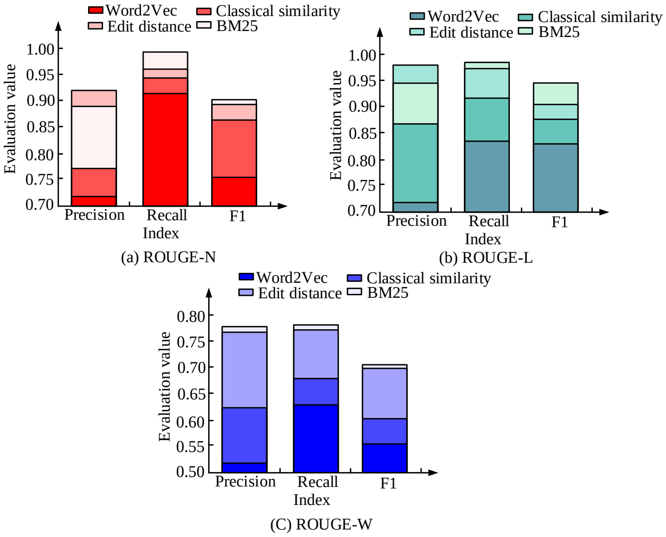
III. Results of the Language Keyword Acquisition Model based on Textrank Algorithm

In order to verify the superiority of the research constructed model, the similarity algorithm and TextRank algorithm chosen for the research are tested for performance and application comparison study, and then analyzed for the actual application of the research constructed model and compared with other methods in the process.

A. *Comparison Results of Similarity Algorithms*

In order to reduce the experimental error, the experiment was analyzed and studied using the same device with Intel Xeon W-2295 CPU, 16G RAM, 100G hard disk memory, Red Hat Enterprise Linux 8 as the operating system, and Python 3.9 as the programming language. The dataset test was obtained from the LC library of students from an elementary school and a secondary school. To test the BM25 similarity algorithm chosen for the study, comparison methods were chosen: classical similarity, edit distance, Word2Vec. these methods were compared with recall-orientated understudy for gisting evaluation (ROUGE) of the BM25 similarity algorithm of the study method. And the average of 100 texts analyzed is shown in Fig. 8.

In Fig. 8, ROUGE is scored in three dimensions, ROUGE-N, ROUGE-L, and ROUGE-W. Three evaluation metrics are selected in each dimension: accuracy, recall with F1 value. For ROUGE-N, Word2Vec has the lowest evaluated values of accuracy, recall and F1 value, and BM25 similarity algorithm with edit distance has comparable evaluated values of accuracy, recall and F1 value. For ROUGE-N, Word2Vec still performs the worst, and the BM25 similarity algorithm with edit distance has a higher recall and a smaller difference in accuracy from the F1 value. For the dimension ROUGE-W the evaluation values are similar to the first two dimensions. It is proved that BM25 algorithm has better performance than other algorithms in various dimensions of ROUGE scoring, especially in the recall rate advantage is crucial to ensure the integrity of automatic summarization. In addition, BM25 algorithm also has better performance on ROUGE score than the conclusion in study [7]. Continuing with the comparison of the processing time of these four methods, it is shown in Fig. 9.



(a) ROUGE-N

(b) ROUGE-L

(C) ROUGE-W

Fig. 8. Rouge scoring results of four methods.

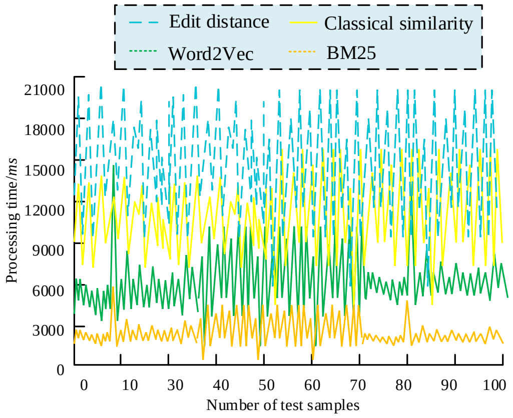


Fig. 9. Comparison of processing time of four methods.

In Fig. 9, in 100 runs, the processing time of the research method is found to be significantly shorter, with an average of only 2182 ms. the average processing time of Word2Vec and classical similarity is 6192 ms and 18065 ms, respectively. The edit distance method mentioned in Fig. 8, although the accuracy, recall and F1 value are not much different from that of the research method, the processing time of the research method is significantly higher, with an average of 20972ms. This means that the processing time of the edit distance method is 8.61 times higher than that of the research method, indicating that the research method has a significant advantage in efficiency. Processing time is one of the key indicators of the usefulness of the algorithm, and the BM25 algorithm shows a shorter processing time, indicating a significant efficiency advantage, suitable for real-time or large-scale text processing tasks.

B. Application of a Language Keyword Acquisition Model Based on Textrank Algorithm

To verify the superiority of TextRank algorithm, it is compared with TF-IDF, LDA and LexRank algorithms. In terms of parameter configuration, the research set 0.90 momentum, 0.0004 attenuation, and planned to conduct 300 rounds of training. The initial learning rate is set to 0.01, and the cosine learning rate strategy is adopted, and the learning rate will be adjusted to 0.001 as the training progresses. Selecting the indicator Receiver operating characteristic curve (ROC) curve, the text in the dataset is tested and analyzed several times, and the comparison results after 50 times are shown in Fig. 10.

In Fig. 10, the area under the ROC curve (AUC) is between 0.1 and 1, providing a direct way to measure the accuracy of the model, and an increase in the AUC value means that the model's predictive accuracy increases. In this figure, the TextRank algorithm has the largest AUC value, which is significantly higher than the other models, at 0.954, very close to 1. This is followed by the TF-IDF algorithm, which also has a higher accuracy with an AUC value of 0.842, and the rest of the models have an AUC value of around 0.70. AUC is an important index to measure the prediction accuracy of the model, and the AUC value of the TextRank algorithm is the largest, close to 1, indicating that its prediction accuracy in the automatic summary task is very high. The results illustrate the superiority of the accuracy of the research method, and continue to compare it with the three methods mentioned above by analyzing 100 texts, and the results of the comparison of accuracy, recall, and F1 value are shown in Fig. 11.

In Fig. 11 (a), the accuracy curve of LexRank has the largest fluctuation, the accuracy is not up to 75%, the range of values of TF-IDF and LDA accuracy is between 75% and 90%, and the accuracy of TextRank algorithm is above 90%, and the accuracy change for 100 text analysis is small, and the curve is relatively flat, with an average accuracy of 94.23%. The comparison between Fig. 11(b) and Fig. 11(c) is similar to that of Fig. 11(a), and the average of the recall and F1 value of TextRank reaches 96.67% and 95.85%, respectively. Accuracy rate, recall rate and F1 value are the key indexes to evaluate the performance of automatic summarization algorithm. TextRank algorithm performs better than other algorithms on these indexes, which proves its superiority and reliability in automatic summarization task. In addition, TextRank algorithm also has more advantages than references [8] and [9]. In order to verify the applicability of the research constructed model, the indicator keyword coverage is selected for application evaluation, which refers to the degree to which the keywords cover the key content of the original text. Prior to this determine the set of keywords of the applied texts, which were selected by manual methods. Ten LCs from different grades were analyzed and the results are shown in Fig. 12.

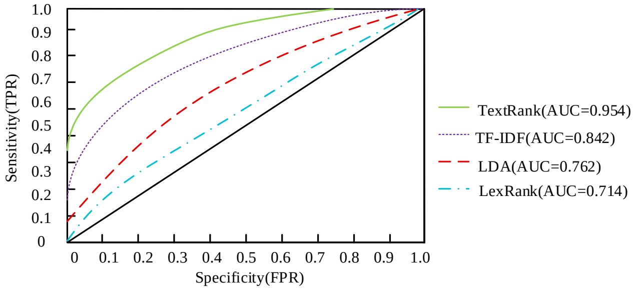


Fig. 10. Comparison of ROC curves for four methods.

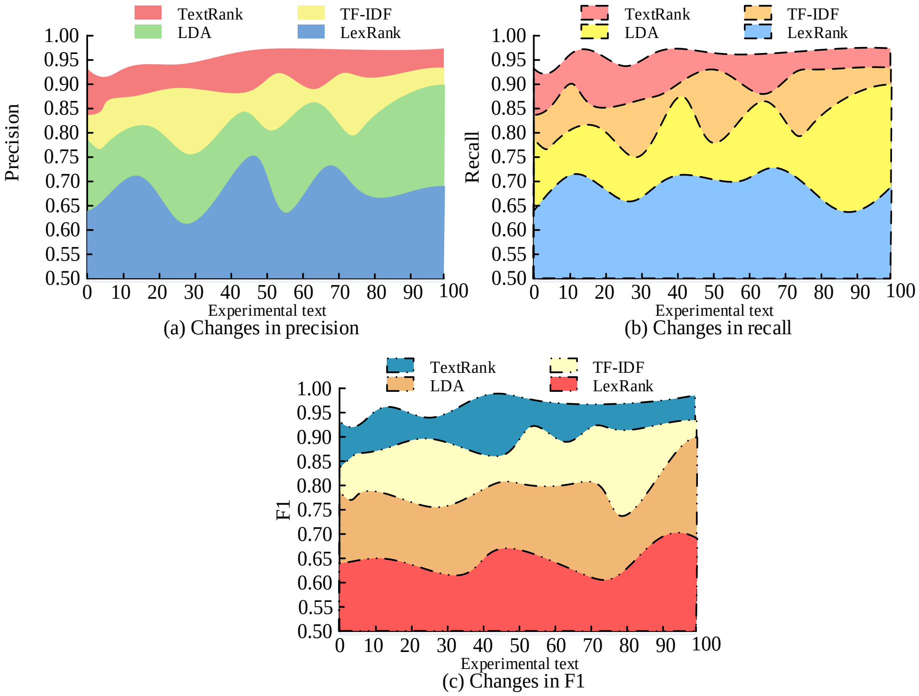


Fig. 11. Comparison results of precision, recall and F1 value of four methods.

In Fig. 12, the keyword coverage of the research model fluctuates the most among the 10 texts, reaching a maximum value of 80.23% at text 6, and it has an average coverage of 62.87%. The keyword coverage fluctuation curves of LDA and TF-IDF are more gentle, with an average keyword coverage of 66.96% and 77.12%, respectively. The curve of the research method has the smallest fluctuation and reaches the maximum value of 96.76% at text 6, with an average keyword coverage of 94.54%. Keyword coverage reflects the comprehensiveness of the algorithm to capture the main information of the original text. The research model performs better than other algorithms in keyword coverage, indicating that it can capture and extract key information of text more comprehensively. For the large gap in the extraction accuracy of the 10 text keywords, it may be due to the fact that the texts selected for the experiment were from different grades. In order to further explore the superiority of the research model, it was continued to be compared with the three methods mentioned above and applied to five LCs, and the scores of classification accuracy, entity accuracy and key description accuracy are shown in Table I.

In Table I, for the comparison of classification accuracy, entity accuracy and key description accuracy for the four methods, classification accuracy could not be compared and only the research methods were able to classify. The difference in entity accuracy was not significant and key description accuracy was greater. By comparing the mean of the sum of the scores, it can be seen that the research model has the highest sum of scores, 3.488, and the remaining three types of models do not have scores higher than 3. The research continues to analyze the text for different grade levels, and the results of its comparison are shown in Fig. 13.

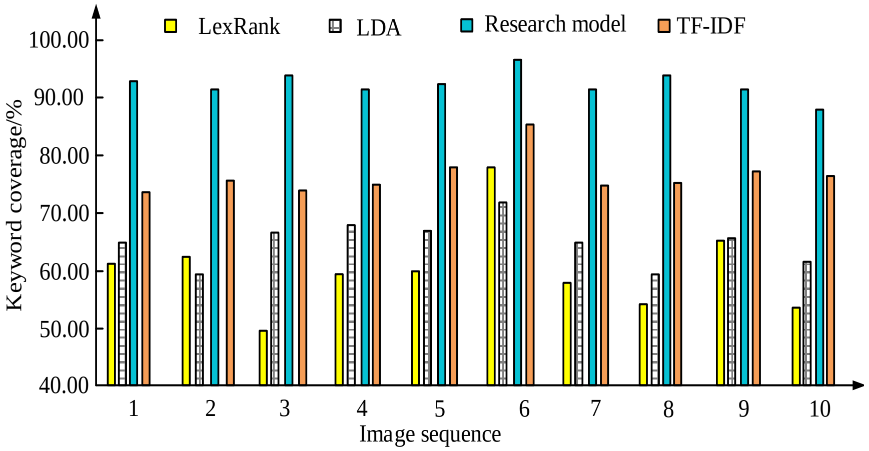
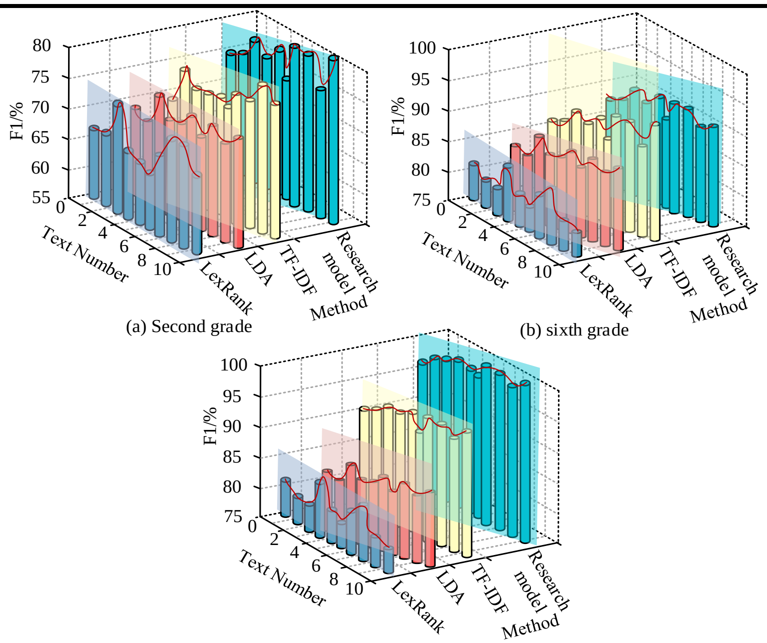


Fig. 12. Comparison of keyword coverage among four methods.

TABLE I. SCORE OF FOUR METHODS FOR CLASSIFICATION ACCURACY, ENTITY ACCURACY, AND KEY DESCRIPTION ACCURACY

Algorithm	Text Number	Classification accuracy	Entity accuracy	Accuracy of key descriptions	Total score	Average value
Research model	1	0.81	1.76	0.95	3.52	3.488
	2	0.80	1.73	0.92	3.45	
	3	0.83	1.75	0.91	3.49	
	4	0.81	1.77	0.95	3.53	
	5	0.82	1.72	0.91	3.45	
TF-IDF	1	-	1.73	0.63	2.36	2.366
	2	-	1.72	0.64	2.36	
	3	-	1.74	0.65	2.39	
	4	-	1.71	0.68	2.39	
	5	-	1.72	0.61	2.33	
LDA	1	-	1.71	0.70	2.41	2.436
	2	-	1.74	0.72	2.46	
	3	-	1.73	0.71	2.44	
	4	-	1.71	0.72	2.43	
	5	-	1.69	0.75	2.44	
TextRank	1	-	1.54	0.51	2.05	2.098
	2	-	1.52	0.57	2.09	
	3	-	1.61	0.53	2.14	
	4	-	1.54	0.54	2.08	
	5	-	1.57	0.56	2.13	



(a) Second grade

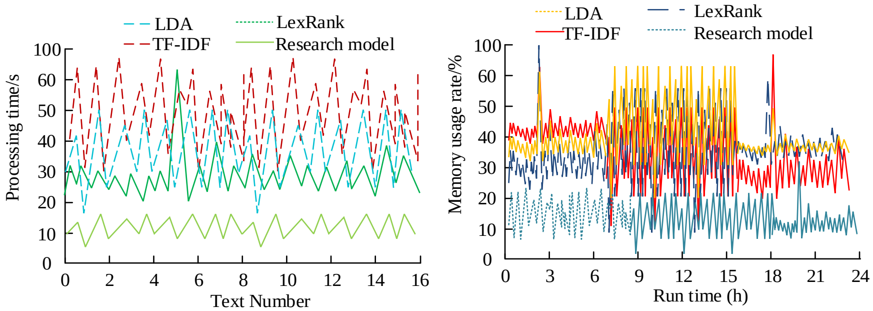
(b) sixth grade

(c) Second grade

Fig. 13. Analysis of four methods for texts in different grades.

Fig. 13 (a) demonstrates the results of analyzing the essays of the second grade, the F1 value of the research method is relatively high, but the difference between the four methods is not significant, the average value of the F1 value of the research method is 77.24%, which does not reach 80%. Fig. 13 (b) demonstrates the results of analyzing the essays of the sixth grade, and the F1 value of the research methods remained higher at 86.94%. Fig. 13 (c) demonstrates the results of analyzing the essays of the ninth grade, the research method has the highest F1 value and it is significantly different from the F1 value of the remaining three methods, the research method F1 value reaches more than 90% and the average F1 value is 96.23%. F1 value is a performance indicator that takes into account accuracy and recall rate. With the improvement of the research method, the F1 value also increases, indicating that the method has better analysis ability for more logical and complex texts, and the F1 value of the research method is always the highest in the comparison of the four methods. Continuing to compare the processing time and memory usage of the four methods, the results are shown in Fig. 14.

In Fig. 14(a), the presented data clearly reveals the significant differences in processing time among the four different methods. Among them, the model used in the study shows the best time efficiency, with its processing time fluctuating mainly around 10 seconds and an average processing time of 11.29 seconds. In contrast, the LexRank, LDA, and TF-IDF methods have longer processing times and show varying degrees of volatility. In addition, Fig. 14(b) provides a comparison of these methods in terms of memory occupancy. In this figure, the research model also shows a significant advantage in terms of memory occupancy, with an average memory occupancy of only 15.67%, whereas the memory occupancy of the other three methods shows greater volatility and instability, with the highest of them even reaching 100%. The processing time and memory usage are directly related to the practicability and scalability of the algorithm. The research model shows advantages in both aspects, which means that it is more suitable for practical application in terms of resource consumption and time efficiency.



(a) Comparison of processing time for four methods

(b) Comparison of Memory Occupancy of Four Methods

Fig. 14. Comparison of processing time and memory usage of four methods.

IV. Conclusion

BM25 similarity algorithm and TextRank algorithm are introduced to obtain keywords in Chinese composition more conveniently. On this basis, a keyword acquisition model is constructed by combining with other intelligent methods. In terms of theoretical contribution, the research expands the application scope of TextRank algorithm in Chinese text processing, and lays a foundation for building a more intelligent education assistance system by optimizing the keyword extraction process, which further promotes the development of AI in the field of education technology. In the actual contribution, the research not only improves the efficiency and accuracy of keyword extraction, reduces the work burden of teachers, but also enhances the objectivity and consistency of evaluation, which has far-reaching research significance for improving teaching efficiency, promoting technological innovation and leading the development of educational technology.

In practical applications, the proposed method shows high accuracy, recall rate and F1 value, and has significant advantages in processing time. It has a short running time, and the average running time is only 2182ms in 100 processing times, and the editing distance processing method is 8.61 times of it. In the performance test and comparison of TextRank algorithm, the AUC value of TextRank algorithm is the largest, which is significantly higher than other models, reaching 0.954, which is very close to 1. The accuracy of TextRank algorithm is above 90%, and the average accuracy of 100 text analysis is 94.23%, and the average recall rate and F1 value reach 96.67% and 95.85% respectively. In the application comparison of the four methods, the research model reaches the maximum value at text 6, which is 96.76%, and the average keyword coverage is 94.54%. For the experimental samples of different grades, the average F1 value of the research model in the second grade was 77.24%, which did not reach 80%. The average F1 value of the model in grade 6 was 86.94%. The average F1 value of the study model at grade 9 was 96.23%. This shows that the accuracy rating value increases with the increase of grade level. In the comparison of the processing time and memory usage of the four methods, the research model shows obvious advantages. The average processing time of 16 texts is 11.29 seconds, and the average memory usage within 24 hours is only 15.67%, which is lower than the other three methods.

Although the research has achieved remarkable results in the accuracy and efficiency of keyword extraction, there are still some limitations. Since the research model is aimed at more complex logically complete utterance, and the selected training corpus is also composed of high-level sentences, the analysis accuracy of LC in lower grades is lower.

Future studies need to select lower-grade LC corpora for training to improve the applicability and practicability of the model. It should also be considered to train the model using a broader corpus of different grades and different types (e.g., argumentative essays, narrative essays, etc.) to improve the applicability of the model to different educational levels and text types. At the same time, the scalability and adaptability of the algorithm are considered, so that it can handle larger scale text data. The methods and techniques studied can be extended to other fields, such as automatic summary generation of legal, medical and scientific documents.

REFERENCES

[1] N. Klyuchnikov and I. Trofimov. "NAS-Bench-NLP: Neural Architecture Search Benchmark for Natural Language Processing," IEEE Access, vol.10, pp. 45736-45747, January 2022, DOI: 10.1109/ACCESS.2022.3169897.

[2] R. Y. Lee, L. C. Brumback, W. B. Lober, J. Sibley, E. L. Nielsen, P. D. Treece, and J. R. Curtis. "Identifying Goals of Care Conversations in the Electronic Health Record Using Natural Language Processing and Machine Learning," J. Pain Symptom Manage., vol. 61, no. 1, pp. 136-142.e2, January, 2021, DOI: 10.1016/j.jpainsymman.2020.08.024.

[3] L. Zhao, W. Alhoshan, A. Ferrari, K. J. Letsholo, M. A. Ajagbe, E.-V. Chioasca, and R. T. Batista-Navarro. "Natural Language Processing for Requirements Engineering," ACM Comput. Surv., vol. 54, no. 3, pp. 1-41, April, 2022, DOI: 10.1145/3444689.

[4] U. Rani and K. Bidhan. "Comparative Assessment of Extractive Summarization: TextRank, TF-IDF and LDA," J. Sci. Res., vol. 65, no. 01, pp. 304-311, January, 2021, DOI: 10.37398/jsr.2021.650140.

[5] M. A. Zamzam. "SISTEM AUTOMATIC TEXT SUMMARIZATION MENGGUNAKAN ALGORITMA TEXTRANK," MATICS, vol. 12, no. 2, pp. 111-116, March, 2021, DOI: 10.18860/mat.v12i2.8372.

[6] K. E. Dewi and N. I. Widiastuti. "The Design of Automatic Summarization of Indonesian Texts Using a Hybrid Approach," J. Teknol. Inf. Pendidik., vol. 15, no. 1, pp. 37-43, November, 2022, DOI: 10.24036/jtip.v15i1.451.

[7] H. Aliakbarpour, M. T. Manzuri, and A. M. Rahmani. "Improving the Readability and Saliency of Abstractive Text Summarization Using Combination of Deep Neural Networks Equipped with Auxiliary Attention Mechanism," J. Supercomput., vol. 78, no. 2, pp. 2528-2555, February, 2022, DOI: 10.1007/s11227-021-03950-x.

[8] Y. Huang, Z. Yu, J. Guo, Y. Xiang, and Y. Xian. "Element Graph-Augmented Abstractive Summarization for Legal Public Opinion News with Graph Transformer," Neurocomputing, vol. 460, pp. 166-180, October, 14, 2021, DOI: 10.1016/j.neucom.2021.07.013.

[9] A. Zagar and M. Robnik-Sikonja. "Cross-lingual transfer of abstractive summarizer to less-resource language," J. Intell. Inf. Syst., vol. 58, no. 1, pp. 153-173, February, 2022. DOI: 10.1007/s10844-021-00663-8.

[10] E. Inan. "Somun: Entity-Centric Summarization Incorporating Pre-Trained Language Models. " Neural Comput. Appl., vol. 33, no. 10, pp. 5301-5311, May, 2021, DOI: 10.1007/s00521-020-05319-2.

[11] D. Qiu and Q. Zheng. "Improving TextRank Algorithm for Automatic Keyword Extraction with Tolerance Rough Set," Int. J. Fuzzy Syst., vol. 24, no. 3, pp. 1332-1342, April, 2022, DOI: 10.1007/s40815-021-01190-y.

[12] Y. F. Hernawan, P. P. Adikara, and R. C. Wihandika. "Peringkasan Artikel Berbahasa Indonesia Menggunakan TextRank Dengan Pembobotan BM25," J. Teknol. Inf. Ilmu Komput., vol. 9, no. 1, pp. 61-68, December, 2022, DOI: 10.25126/jtiik.2022913765.

[13] Z. Huang and Z. Xie. "A Patent Keywords Extraction Method Using TextRank Model with Prior Public Knowledge," Complex Intell. Syst., vol. 8, no. 1, pp. 1-12, February, 2022, DOI: 10.1007/s40747-021-00343-8.

[14] M. F. Fakhrezi, Moch. A. Bijaksana, and A. F. Huda. "Implementation of Automatic Text Summarization with TextRank Method in the Development of Al-Qur’an Vocabulary Encyclopedia," Procedia Comput. Sci., vol. 179, pp. 391-398, January, 2021, DOI: 10.1016/j.procs.2021.01.021.

[15] S. Zhang, Q. Luo, Y. Feng, K. Ding, D. Gifu, S. Zhang, and J. Xia. "Key Phrase Extraction by Improving TextRank with an Integration of Word Embedding and Syntactic Information," Recent Adv. Comput. Sci. Commun., vol. 14, no. 9, pp. 2969-2975, December, 2021, DOI: 10.2174/2666255813999200820155846.

[16] A. B. K. Susanto, N. Muliadi, B. Nugroho, and M. Muljono. "Comparison of String Similarity Algorithm in Post-Processing OCR," J. Appl. Intell. Syst., vol. 8, no. 1, pp. 25-32, June, 2023, DOI: 10.33633/jais.v8i1.7079.

[17] A. Kurnianti, P. Pahlevi, and I. Mufidah. "Recommendation System for Prospective Bride and Groom Using Cosine Similarity Algorithm," Emerg. Inf. Sci. Technol., vol. 4, no. 1, pp. 8-15, June, 2023, DOI: 10.18196/eist.v4i1.18683.

[18] J. S. Baruni and Dr. J. G. R. Sathiaseelan. "Keyphrase Extraction from Document Using RAKE and TextRank Algorithms," Int. J. Comput. Sci. Mob. Comput., vol. 9, no. 9, pp. 83-93, October, 2020, DOI: 10.47760/ijcsmc.2020.v09i09.009.

[19] P. Preethi and H. R. Mamatha. "Region-Based Convolutional Neural Network for Segmenting Text in Epigraphical Images," Artif. Intell. Appl., vol. 1, no. 2, pp. 119-127, September, 2023, DOI: 10.47852/bonviewAIA2202293.

[20] H. Mokayed, T. Z. Quan, L. Alkhaled, and V. Sivakumar. "Real-time human detection and counting system using deep learning computer vision techniques," Artif. Intell. Appl., vol. 1, no. 4, pp. 221-229, September, 2023, DOI: 10.1109/ViTECoN58111.2023.10157694.

[21] K. Zheng, Y. Li, and W. Xu. "Regime Switching Model Estimation: Spectral Clustering Hidden Markov Model," Ann. Oper. Res., vol. 303, no. 1-2, pp. 297-319, August, 2021, DOI: 10.1007/s10479-019-03140-2.

[22] X. Wan, T. Han, J. An, and M. Wu. "Hidden Markov Model Based Fault Detection for Networked Singularly Perturbed Systems," IEEE Trans. Syst., Man, Cybern.: Syst., vol. 51, no. 10, pp. 6445-6456, October, 2021, DOI: 10.1109/TSMC.2019.2961978.

[23] Y. Li, E. Zio, and E. Pan. "An MEWMA-Based Segmental Multivariate Hidden Markov Model for Degradation Assessment and Prediction," Proc. Inst. Mech. Eng., Part O: J. Risk Reliab., vol. 235, no. 5, pp. 831-844, October, 2021, DOI: 10.1177/1748006X21990527.

[24] W. Zhao, T. Shi, and L. Wang. "Fault Diagnosis and Prognosis of Bearing Based on Hidden Markov Model with Multi-Features," Appl. Math. Nonlinear Sci., vol. 5, no. 1, pp. 71-84, January, 2020, DOI: 10.2478/amns.2020.1.00008.

[25] Y. Lu and S. An. "Research on Sports Video Detection Technology Motion 3D Reconstruction Based on Hidden Markov Model," Cluster Comput., vol. 23, no. 3, pp. 1899-1909, September, 2020, DOI: 10.1007/s10586-020-03097-z.

[26] S. Dong, Z.-G. Wu, P. Shi, H. Su, and T. Huang. "Quantized Control of Markov Jump Nonlinear Systems Based on Fuzzy Hidden Markov Model," IEEE Trans. Cybern., vol. 49, no. 7, pp. 2420-2430, July, 2019, DOI: 10.1109/TCYB.2018.2813279.

[27] L. Shen, X. Yang, J. Wang, and J. Xia. "Passive Gain-Scheduling Filtering for Jumping Linear Parameter Varying Systems with Fading Channels Based on the Hidden Markov Model," Proc. Inst. Mech. Eng., Part I: J. Syst. Control Eng., vol. 233, no. 1, pp. 67-79, January, 2019, DOI: 10.1177/0959651818777679.

[28] R. A. Pratama, A. A. Suryani, and W. Maharani. "Part of Speech Tagging for Javanese Language with Hidden Markov Model," J. Comput. Sci. Inf. Eng. (J-Cosine), vol. 4, no. 1, pp. 84-91, July, 2020, DOI: 10.29303/jcosine.v4i1.346.

[29] V. Sorin, Y. Barash, E. Konen, and E. Klang. "Deep Learning for Natural Language Processing in Radiology—Fundamentals and a Systematic Review," J. Am. Coll. Radiol., vol. 17, no. 5, pp. 639-648, May, 2020, DOI: 10.1016/j.jacr.2019.12.026.

[30] Q. Zhao, J. Niu, and X. Liu. "ALS-MRS: Incorporating Aspect-Level Sentiment for Abstractive Multi-Review Summarization," Knowl.-Based Syst., vol. 258, pp. 1-14, December, 2022, DOI: 10.1016/j.knosys.2022.109942.