IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 June 2023, accepted 3 July 2023, date of publication 17 July 2023, date of current version 14 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3296141

RESEARCH ARTICLE

Application of Text Rank Algorithm Fused With LDA in Information Extraction Model

YUNBO WEI AND YONGSHENG DING
School of Sciences, Qiqihar University, Qiqihar 161006, China

Corresponding author: Yunbo Wei (01027@qqhru.edu.cn)

This work was supported in part by the Scientific Research Projects of Heilongjiang Provincial Education Department—Research on Multi-Block Image Encryption Based on Cascade Chaotic Mapping under Grant 135509313, in part by the Natural Science Foundation of Heilongjiang Province of China—Statistical Inference of Stochastic Environmental Branching Processes Based on High-Dimensional Data under Grant LH2019A027, in part by the Higher Education Reform of Heilongjiang Province of China—Research on the Strategy of the Continuous Reform of Real-Time Online Interactive Courses in the Post-Epidemic Era under Grant SJGY20200772, and in part by the Basic Scientific Research Expenses of Colleges and Universities in Heilongjiang Province under Grant 145209138.

ABSTRACT With the rapid development of network technology, a large amount of information fills the network world, and the performance of the current information extraction model to extract keyword information from a large number of data is insufficient. To solve the problem of insufficient extraction performance in traditional information extraction models, this paper combines text sorting algorithms with document topic generation models. A keyword information extraction model that combines the advantages of the two algorithms is proposed. The performance comparison experiment of this fusion algorithm shows that its accuracy and recall rates are 76.1% and 77.0%, respectively, which outperform the comparing algorithm. In the empirical analysis results of the information extraction model, it is found that the accuracy and precision rates of the proposed information extraction model are 80.16% and 77.54%, respectively, which are better than the comparing model. The proposed model of information extraction is of great importance for the development of the field of information extraction.

INDEX TERMS LDA, TextRank, information extraction model, keyword information, weight.

I. INTRODUCTION

With the development of the Internet, a huge amount of network information has been generated. Information extraction technology (IET) has been extensively used in the field of language processing [1], [2]. IET is the extraction of target information from network text, which is mainly divided into three categories: information extraction, classification and disambiguation, and named entity recognition [3], [4]. Information extraction is the process of extracting information from entities in network texts for the purpose of determining the structure and content of the target texts [5]. Information extraction occupies an essential position in the field of natural language processing. It can directly address the problem of information overload. It is a highly demanding and significant work to mankind [6]. In Cai et al.'s research, it was mentioned that there are currently two main information extraction technologies: Chinese word segmentation technology and text classification technology. Chinese word segmentation technology mainly extracts information through Chinese word segmentation, while text classification technology uses various algorithms to classify and extract text [7]. These methods based on statistical models and machine learning algorithms have become the most mainstream methods in IET due to their strong interpretability and good generalization ability [8]. However, these traditional methods have certain limitations [9]. For example, the traditional TextRank algorithm has the problems of non-intuitive feature representation and high computational complexity when processing large text data, so it usually needs to construct the feature representation matrix manually. In view of this, this paper introduces a Latent Dirichlet Allocation (LDA) and combines traditional text vector representation methods with machine learning algorithms to design an information extraction model (IEM) built on the fusion of LDA and TextRank algorithms. The fusion algorithm is obtained by fusion of LDA and TextRank,

The associate editor coordinating the review of this manuscript and approving it for publication was Adnan Abid.

and the information extraction model is constructed on this basis. The information extraction model is applied to actual information extraction, so as to improve the accuracy of the current information extraction model and provide a new idea for obtaining more valuable information in the era of big data. The contribution of this study is to organically combine LDA model and TextRank algorithm, and this model can improve the accuracy of information extraction and promote the development of information extraction field. This paper is divided into four parts, the first part mainly introduces the TextRank algorithm and information extraction model related research; The second part is the construction and application of information extraction model based on LDA-TextRank algorithm. The third part is the performance comparison of the fusion algorithm and the performance test of the information extraction model. The fourth part is the conclusion.

II. RELATED WORK

As a common keyword extraction algorithm, TextRank algorithm and its improved algorithm have been widely used in translation, news and other fields [10], [11]. To improve the extraction performance of traditional TextRank algorithms, Zhang et al. proposed a word embedding and syntactic information algorithm that considers syntactic and semantic information. Through performance testing of the algorithm, it is concluded that the performance of the unsupervised keyword extraction algorithm is superior to the traditional TextRank algorithm [12]. Due to the difficulty in understanding the vocabulary of the Quran, Fakhrezi's team had launched an encyclopedia of Quranic vocabulary based on the TextRank algorithm. The improved TextRank algorithm was tested and obtained an F-value (harmonic mean of accuracy and recall) of 0.6173. The results of automatic text summarization using this algorithm would not be repeated, which has important practical value [13]. To solve the problem of low accuracy of current keyword extraction algorithms, Bordoloi et al. proposed a TextRank algorithm based on a unique statistical supervision weight. The performance comparison experiment of the proposed algorithm showed that the accuracy of the algorithm was 73.2%, which was better than 68.5% and 66.7% of the comparison algorithm. The results show that the proposed algorithm can effectively improve the accuracy of keyword extraction algorithm and has practical significance [14]. To better extract the commercial value of various perspectives on social media, Jun's team demonstrated the TextRank model of Word2vec and Doc2vec. This model adjusted the weight of the generated keywords by calculating the jump probability between nodes, and ultimately sorted the generated keywords. According to empirical analysis, the model had good extraction performance in various datasets [15].

With the fast-developed network technology, a mass of information is flooding the network world, so the requirements for extracting key information from network information are gradually increasing. Therefore, there are various advanced technologies are applied in IEM. Zhou et al. put forward an IEM on the basis of free-text clinical IET in order to provide more accurate information for personalized medicine. They validated the performance of the model through simulation and the use of clinical records. The results show that using the information extracted by this model can provide better targeted treatment for patients and improve the cure rate [16]. Martinez et al. proposed an information extraction strategy according to Semantic Web standards to address the problem that current product recommendation systems cannot provide targeted recommendations. The strategy included information extraction tasks and mixed semantic similarity measures. After analysis, it was found that its information extraction performance is superior, which can improve the recommendation accuracy of the system and promote reasonable consumption by people [17]. Sun and Ren raised a fault diagnosis algorithm to improve its accuracy in accordance with multi-scale feature extraction. The algorithm utilized the Gramian angular field to fully extract the temporal information from the data. The recall rate and F-value obtained through experimental comparison tests were superior to the comparison algorithm [18]. To improve the performance of beamforming methods, Liu and other researchers explored downlink beamforming designs in line with spatial and physical channel information extraction. Compared to the most advanced beamforming methods, the algorithm's performance was significantly improved [19]. Steinkamp et al. put forward an IEM on the basis of machine learning to fully utilize the information in the clinical informatics information repository. Empirical analysis showed that the information extraction accuracy of the model was superior to that of the comparative model [20].

From the perspective of the algorithm, the above research analyzes the status quo that TextRank algorithm, as a common keyword extraction algorithm, has been widely used in many fields, and the accuracy of the algorithm extraction can be effectively improved by improving the algorithm. In addition, the above content also illustrates that in the field of information extraction, many advanced technologies have been applied in the information extraction model, by taking advantage of advanced technologies to improve the performance of the information extraction model. Therefore, this paper proposes an improved TextRank algorithm based on the document topic generation model, and applies the new algorithm to the information extraction model. This method can improve the problem that the accuracy of information extracted by the traditional information extraction model is not high, and provide a new idea and certain data support for the organic integration of algorithm domain and information extraction domain. This method is expected to improve the problem that the accuracy of information extraction is not high in the traditional information extraction model, and provide a new idea and certain data support for the organic integration of algorithm and information extraction.

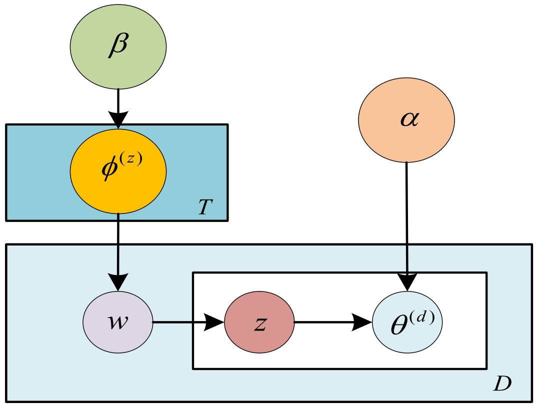


FIGURE 1. LDA topic model architecture.

III. CONSTRUCTION AND APPLICATION OF AN IEM FOR LDA TextRank ALGORITHM

A. IEM CONSTRUCTION BASED ON LDA ALGORITHM

LDA topic model is a probabilistic analysis method that incorporates implicit semantics built on Bayesian theory [21]. Based on the LDA model, the LDA algorithm introduces the view of the Bayes school, in which the word distribution of the topic and the topic distribution of the document will change, and uses hyperparameters to generate the word distribution of the topic and the topic distribution of the document [22]. Figure 1 shows the architecture of the LDA topic model. The specific steps of the LDA model building process are divided into four steps. First, the specific text parameter α in the Dirichlet distribution is used for topic modeling. The Dirichlet distribution is a probability distribution that describes the probability of occurrence of random variables on each number. Second, a topic is randomly selected from the topic distribution of the file generated above. The parameter β in the Dirichlet distribution is then sampled and word distributions on the topic are generated. Finally, a word is randomly selected from the distribution of subject words obtained in the previous step, and a new document is generated using that word. Using the above steps, a word in a new document using the LDA model can be generated.

The LDA algorithmin Figure 1, the topic distribution of the document and the word distribution of the topic satisfy the Dirichlet distribution, and the hyperparameters are α and β, respectively. Supposing there are M documents in total, including K topics. The files in each topic have a corresponding topic distribution, and each topic has a multiple distribution. For each word in a document, it is first extracted from a topic in the document's topic distribution and represented by an integer index. For each word in a document, it is first extracted from a topic in the topic distribution of the document and represented by an integer index. Then, a word is extracted from the topic word distribution corresponding to the topic, and the cycle continues until word extraction is completed in all M documents. In Figure 1, ϕ and θ are the document topic and topic word probability distribution. α and β are hyperparameters of ϕ and θ. w represents a word; z represents the topic distribution of words. According to the LDA graph model in Figure 1, the joint distribution expression of variables shown in Equation (1) can be obtained.

$$p(w_m, z_m, \theta_m, \underline{\Phi} \,|\alpha, \beta) = \prod_{n=1}^{N_m} p(w_{m,n} \,|\varphi_{z_{m,n}})p(z_{m,n} \,|\theta_m) \cdot p(\theta_m \,|\alpha) \cdot p(\underline{\Phi} \,|\beta) \tag{1}$$

N_m in Equation (1) refers to the sum of words in the m-th text, p represents probability. The probability calculation formula for initializing the word $w_{m,n}$ to the word t in the LDA topic model is Equation (2).

$$p(w_{m,n} = t \,|\theta_m, \underline{\Phi}) = \sum_{k=1}^{K} p(w_{m,n} = t \,|\varphi_k)p(z_{m,n} = k \,|\theta_m) \tag{2}$$

The likelihood function expression of the document set in the LDA topic model is Equation (3).

$$p(W \,|\underline{\theta}\ , \underline{\Phi}) = \prod_{m=1}^{M} p(w_m \,|\theta_m, \underline{\Phi}) = \prod_{m=1}^{M} \prod_{n=1}^{N_m} p(w_{m,n} \,|\theta_m, \underline{\Phi}) \tag{3}$$

Equation (4) is the probability distribution expression obtained through Gibbs sampling.

$$\varphi_{k,t} = \frac{n_k^{(t)} + \beta_t}{\sum_{t=1}^{V} n_k^{(t)} + \beta_t} \tag{4}$$

The expression for indirectly estimating θ through Gibbs sampling of variable Z is shown in Equation (5).

$$\theta_{k,t} = \frac{n_m^{(k)} + \alpha_k}{\sum_{k=1}^{K} n_m^{(k)} + \alpha_k} \tag{5}$$

$\theta_{k,t}$ in Equation (5) is the probability of the topic t in d_m of the k-th document. $n_m^{(k)}$ means the times' number that the topic t appears in d_m. In the LDA topic model, hyperparameters are used to generate topic distribution and word distribution, and then a document is generated. The process of generating keywords is as follows. Assuming that there are three topic distributions and each topic word corresponds to three word distributions, when the LDA topic model is used to generate documents, the corresponding topic probability is not the same each time. For example, the LDA algorithm has a selection probability of 0.6, 0.2, and 0.2 for the three different topics the first time it selects them. However, when selecting a topic for the second time, the selection probabilities for the three different topics change to 0.5, 0.3, and 0.2. The LDA algorithm also has different probabilities when selecting topic words in each topic. Figure 2 is the flowchart of keyword generation from the LDA topic model.

Extracting keywords in Figure 2 from the LDA first uses the Dirichlet distribution to generate a corresponding topic distribution, and each time the topic distribution obtained is different. The probability distribution in the topic and the

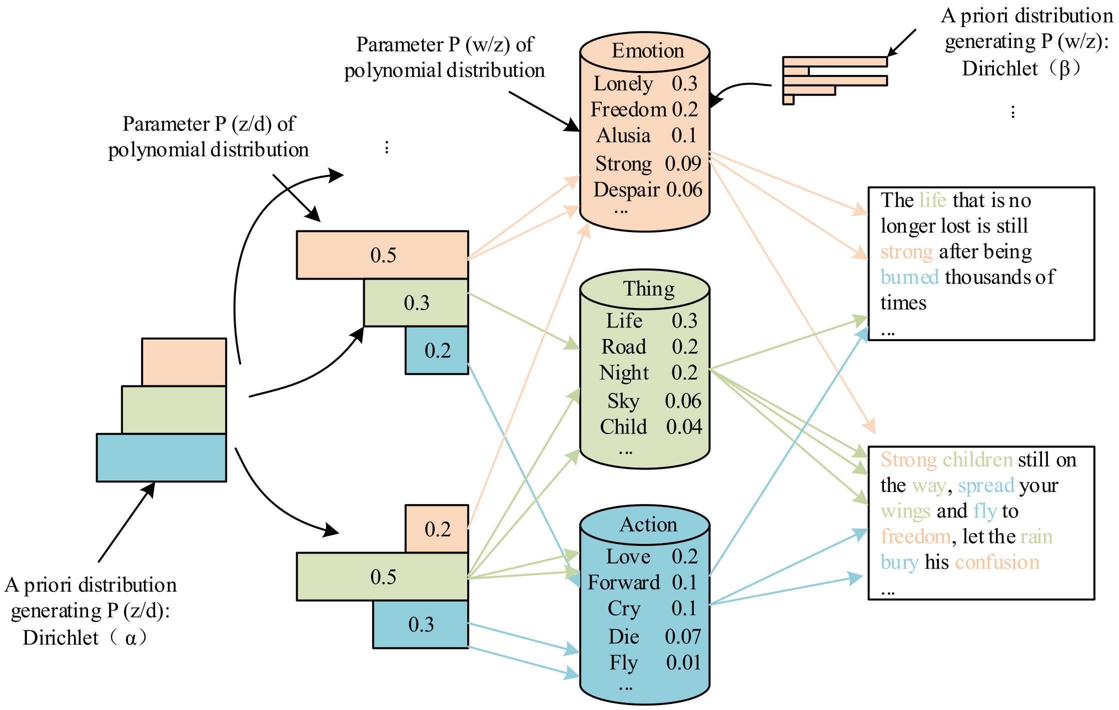

Parameter P (w/z) of polynomial distribution
A priori distribution generating P (w/z): Dirichlet（β）
Parameter P (z/d) of polynomial distribution
Emotion
Lonely 0.3
Freedom 0.2
Alusia 0.1
Strong 0.09
Despair 0.06
...
Thing
Life 0.3
Road 0.2
Night 0.2
Sky 0.06
Child 0.04
...
Action
Love 0.2
Forward 0.1
Cry 0.1
Die 0.07
Fly 0.01
...
0.5
0.3
0.2
0.2
0.5
0.3
The life that is no longer lost is still strong after being burned thousands of times
...
Strong children still on the way, spread your wings and fly to freedom, let the rain bury his confusion
...
A priori distribution generating P (z/d): Dirichlet（α）

FIGURE 2. **The process of generating keywords from the LDA topic model.**

probability distribution of keywords in the topic are variables. The LDA topic model extracts key information from the document-word mapping relationship by adding a hidden topic [23]. In terms of the distribution of words in a topic, if a word occurs more often in the topic, it is considered to be more important to the topic. If a topic occurs frequently in a text, it is considered to make a significant contribution to the document. When extracting information from text, keywords are usually extracted first. The LDA topic extraction model is often applied in keyword extraction because of its good extraction accuracy. The process of keyword extraction model integrated with LDA is shown in Figure 3. The model is mainly divided into steps such as document preprocessing, candidate word selection, candidate word feature calculation, candidate word topic feature calculation, and keyword determination.

Figure 3 is a flowchart of a common keyword extraction model. Among them, document preprocessing refers to the segmentation of sentences and words in a document. Usually, the document is divided into different words and sentences in accordance with different punctuation marks, and then the words in different words and sentences are marked with part of speech. Candidate words are selected from the preprocessed text and fed into the LDA topic model to compute topic features. The candidate word selection process is the foundation of this model. A common document typically contains abundant words, and different words may be combined into massive phrases. If all words and phrases are weighted, the efficiency of the model will be greatly affected. Therefore, before evaluating the weight of words, phrases that can become key words in the whole document are screened, and the selected words are called candidate words. After selecting the candidate words, the weight of the candidate words is determined by calculating the candidate word feature and the candidate word topic feature. The candidate words that meet the document requirements are ultimately determined as the keywords of the entire document [24]. The word feature calculation method is used in the topic feature calculation of candidate words. The statistical characteristics of words generally refer to the location of the first occurrence of words, the length of words, and so on. The term frequency inverse document frequency (TF-IDF) method is a commonly used weighting technique in information retrieval and data mining. The TF-IDF method is used to calculate the statistical features of words, and its statistical feature expression is shown in equation (6).

$$p_{statistic}(d, w_j) = \frac{TF * IDF}{firstOCC} \tag{6}$$

In Equation (6), $p_{statistic}(d, w_j)$ represents the statistical characteristics of the words, and *firstOCC* represents the position where the words first appeared. *TF* indicates word frequency; *IDF* represents the inverse text frequency index. The calculation formula of *IDF* is shown in Equation (7).

$$IDF = \log_2(\frac{D}{\#n(w_j \in d_i)} + 1) \tag{7}$$

In Equation (7), D is the sum of all documents in the corpus.

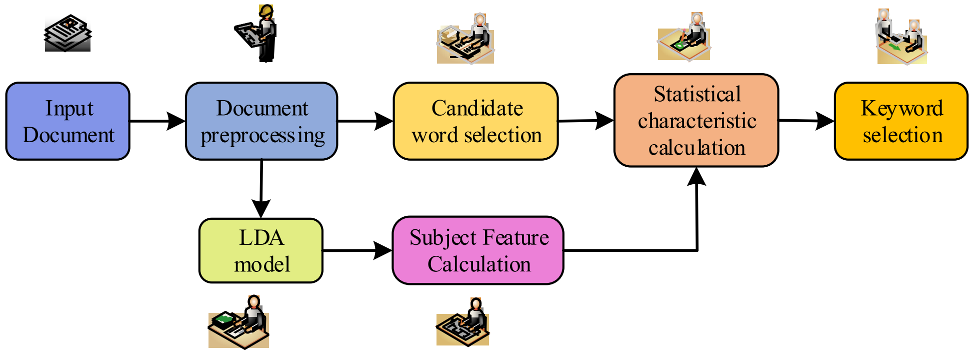


FIGURE 3. Flow chart of information extraction model based on LDA model.

B. OPTIMIZATION OF LDA EXTRACTION MODEL BASED ON TEXTRANK ALGORITHM

Although traditional LDA extraction models can improve the accuracy of key information extraction by adjusting the weights [25]. However, the performance of keyword candidate extraction is weak. As a result, the research combines it with TextRank to increase the extraction accuracy of the extraction model. The TextRank is developed from the page sorting algorithm. The basic idea of this method is to divide text into the smallest units. Using text as a network node, a text graph model is established in accordance with the degree of text co occurrence in the co occurrence window. This is used to express the structural relationship of the text [26]. Then, the graph iteration is used to calculate the importance of these words, thereby judging these words. In the TextRank, the expression formula for the text ranking value of each word node is shown in Equation (8).

$$S(V_i) = (1-d) + d * \sum_{V_i \in \ln(V_i)} \frac{W_{ij}}{\sum_{v_k \in Out(V_j)} W_{jk}} S(V_j) \quad (8)$$

Equation (8) is a recursive formula that uses TextRank to calculate the criticality of words, where d represents the damping coefficient, with a value range of 0-1. The larger the damping coefficient, the greater the iteration numbers, resulting in instability of the algorithm. If the damping coefficient is too small, resulting in an insignificant iterative effect, the damping coefficient is typically set to 0.85. If only text sorting algorithms are used for keyword extraction, the semantic information of words in the entire document is not taken into account. Figure 4 shows the process of filtering candidate keywords using a conventional TextRank algorithm.

V in Figure 4 represents the vertex set; $V_i(i = 1, 2, 3, 4, 5, 6)$ represents the weight of the corresponding word. Because TextRank only considers the dependencies between text terms and does not consider topic as an implicit variable, the keyword information in the document cannot be accurately extracted. Research calculates the topic influence of each word by multiplying the text topic distribution and the topic vocabulary distribution in the LDA topic model, and combines them with text sorting algorithms to obtain a new text extraction model. The research considers topics as intermediate variables to calculate the importance of a word in a document, and the expression is equation (9).

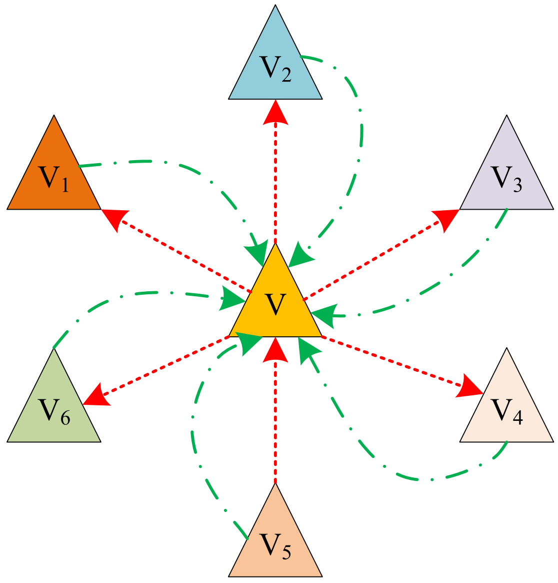


FIGURE 4. The process of selecting candidate keywords using a conventional TextRank algorithm.

$$ti(w\,|d\,) = \sum\nolimits_{j=1}^{t} (\vartheta_Z^{(d)} * \varphi_w^{(z=j)}) \quad (9)$$

In Equation (9), $ti(w\,|d\,)$ is the importance of the word relative to the document. a represents a document. w is the word in the document. In traditional TextRank, the importance of each word is constantly updated and calculated as the formula iterates. To integrate the word influence calculated by the topic model into the text sorting algorithm, research has been conducted to adjust the weight of nodes in the network. The weight adjustment is mainly separated into two parts: one is to adjust the original weight, and the second part is to adjust the topic impact weight obtained by the topic model algorithm. Keyword extraction is the core of the topic model algorithm. The keyword extraction step implemented on the basis of supervised learning can achieve good results, but the cost is too high and does not match the current topic

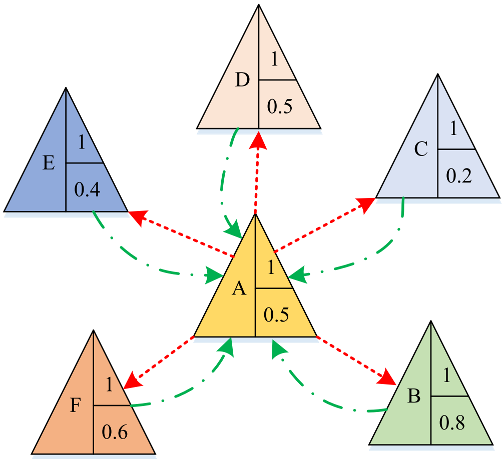


FIGURE 5. A Keyword information extraction model combining LDA algorithm and text sorting algorithm.

model algorithm research. Therefore, the research adopts unsupervised learning to extract keywords from text, and then uses topic model algorithms to calculate the words' weight in the text. Research and then use it to repeatedly count the weight values of each point in the text sorting graph model. Figure 5 shows the keyword IEM, which combines the LDA algorithm and the text sorting algorithm.

Figure 5 shows a keyword extraction model that combines the LDA topic model and TextRank. The extraction model contains six nodes: A, B, C, D, E, and F, with node A as the central node. In traditional keyword extraction methods, the probability of propagation from node A to other nodes is 20%. However, this is not the case in reality, so for improving the accuracy of keyword extraction of the TextRank, research has been conducted to change the weight initialization expression of the TextRank to that shown in Equation (10).

$$TR(v_i) = d\left(\alpha \sum\nolimits_{j:v_j-v_i} \frac{TI(v_i)}{OTI(v_j)} TR(v_j) + \beta \sum\nolimits_{j:v_j-v_i} \frac{1}{OD(v_j)} R(v_j)\right) + (1-d)\frac{1}{|v|} \tag{10}$$

In Equation (10), $TR(v_j)$ is the topic influence weight of the word. $R(v_j)$ indicates that the word affects the weight throughout the document. In Figure 5, the middle node A is divided into two parts. The upper right part represents the traditional TextRank, while the lower right part represents the topic weight of the corresponding word of the node and the document it maps to. The weight of this topic does not change with the number of iterations, and only the weight of the upper right part changes during the iteration process. If a document D contains m keywords and the word graph of the topic model includes m vertices, the initialization weight expressions for all vertices are Equation (11).

$$B_O = \{\frac{1}{n}, \frac{1}{n}, \cdots \frac{1}{n}\} \tag{11}$$

The study constructs probabilistic transitions between words by defining a state transition matrix, whose expression is shown in Equation (12).

$$w = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ w_{31} & w_{32} & \cdots & w_{3n} \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \tag{12}$$

w_{ij} in Equation (12) is the transition probability from vertex v_i to vertex v_j, and the sum of each column is 1. The numerical expression of each element in the matrix of Equation (12) is shown in Equation (13).

$$w_{ij} = \alpha \frac{TI(v_i)}{OTI(v_{ij})} + \beta \frac{1}{OD(v_i)} \tag{13}$$

This method is used to iterate over the weight of each word. After several iterations, the probability transfer matrix of the word subject influence factor is introduced, and the iterative expression is updated to Equation (14).

$$B_i = d^* M^* B_{i-1} + (1-d)^* e/n \tag{14}$$

In Equation (14), B_i is the state transition matrix after iteration i. e represents a vector with a component of 1 and a digit of n. When the difference between the results of two similar iterations is small, separating the structural weights of vector B, and then continue sorting. Output the top ranked words in the remaining parts as key information. To better integrate the benefits of the topic model algorithm and TextRank, the organic integration of the two can be explored. The IEM framework for integrating topic model algorithms and TextRank keywords is shown in Figure 6. The IEM is basically divided into three parts. The first part is to obtain the topic distribution of the text through the topic model. The second mainly constructs a keyword map through the word co-occurrence relationship and similarity relationship of the text, and uses TextRank to obtain candidate keywords. The third part is to iterate through the candidate keywords to obtain the final keyword.

IV. PERFORMANCE COMPARISON RESULT ANALYSIS AND IEM EMPIRICAL ANALYSIS OF FUSION ALGORITHMS

In the research, the Windows 10 operating system is used to conduct comparative experiments to test the performance of the fusion algorithm. The operating system has 8G of memory, 256G of disk space, and contains four cores and eight threads. This study compares the performance of the fusion algorithms with TFIDF, TextRank, and LDA to ensure that all algorithms run in the Python 2.7 environment. The research analyzes the performance of the fusion algorithm using training error, accuracy, F-value, recall rate, and ROC curve as comparison indicators. Figure 7 shows the training error results of the four algorithms.

As Figure 7, the overall training error of the four algorithms shows a downward trend, and the training error curve position of the fusion algorithm is lower than that of the other three algorithms. In addition, the training error of the fusion

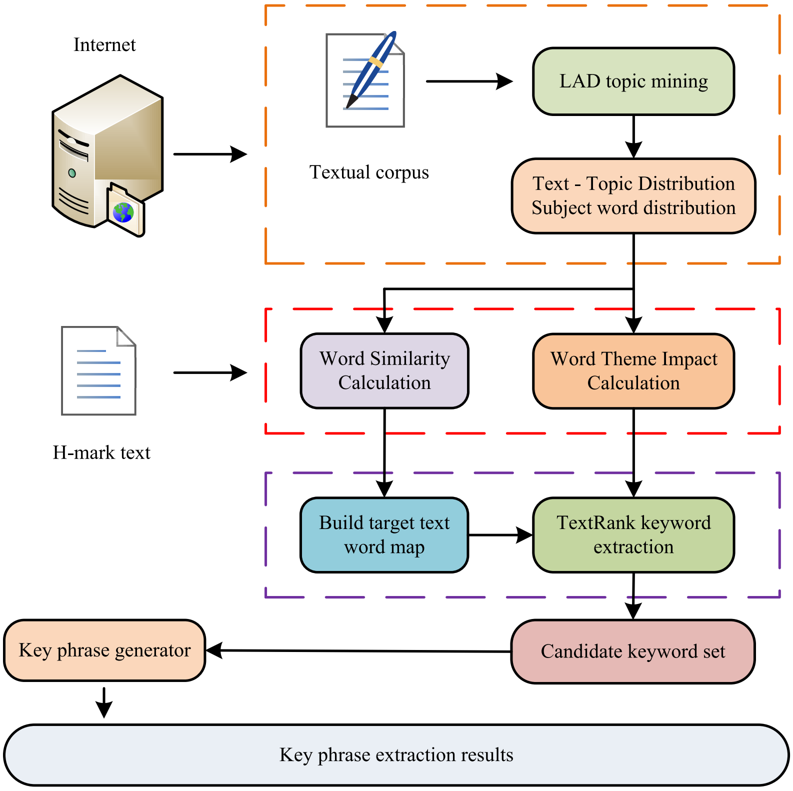

Internet
Textual corpus
LAD topic mining
Text - Topic Distribution
Subject word distribution
H-mark text
Word Similarity Calculation
Word Theme Impact Calculation
Build target text word map
TextRank keyword extraction
Key phrase generator
Candidate keyword set
Key phrase extraction results

FIGURE 6. Overall structure of keyword information extraction model.

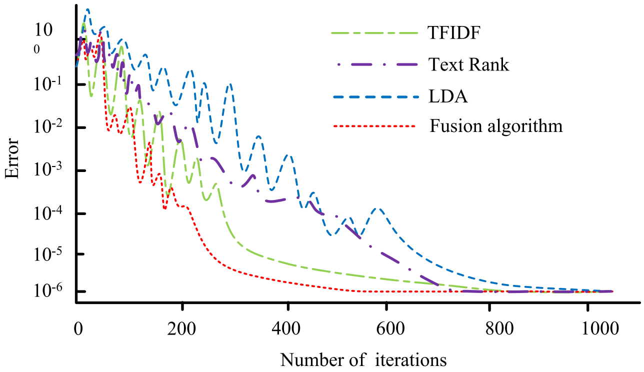


FIGURE 7. Training error results of four algorithms.

algorithm tends to stabilize when the number of iterations is 560, which is earlier than the other three. The above results indicate that the performance of the fusion algorithm is superior to LDA, TFIDF, and Text Rank in terms of training error dimensions. The accuracy results of the four algorithms under different k values are Figure 8. The number of sampled subjects is k.

In Figure 8, except for TextRank, the accuracy rates of the other three algorithms increase with the increase of the k value. Among the four algorithms, the fusion algorithm can reach the highest accuracy of 76.1% when k is about 150, which is higher than the 69.3% of TFIDF algorithm, 64.0% of the TextRank algorithm and 57.9% of the LDA algorithm. The above results prove that from the perspective of accuracy, the fusion algorithm's performance is better than the other three comparative algorithms. In the fusion algorithm, its accuracy increases with the increase of the k value. This is because the larger the number of subjects sampled, the larger the number of samples, the higher the accuracy. The recall rate and F-value results of the four algorithms are Figure 9.

Figure 9 is a comparison diagram of the recall rate and F-value results of the four comparison algorithms. Figure 9(a) is the recall comparison results of the four comparison algorithms. In Figure 9(a), with the increase of the k value, the recall rates of all three algorithms except TextRank show an increasing trend, and the growth trend of the fusion algorithm is the most obvious. The fusion algorithm has the maximum recall rate of 0.77 when the k value is 180. Figure 9(b) shows the F-value comparison results of the four comparison algorithms. From Figure 9(b), as the value of k increases, the F values of all of them show an increasing trend, and the growth trend of the fusion algorithm is the most obvious. The fusion algorithm has a maximum F value of 0.35 when the k value is 180. The above data shows that the performance of the fusion algorithm is more excellent than the other three comparative algorithms from the perspective of recall rate and F-value dimensions. In the fusion algorithm, as the k-value increases, the recall rate and F-value of the algorithm

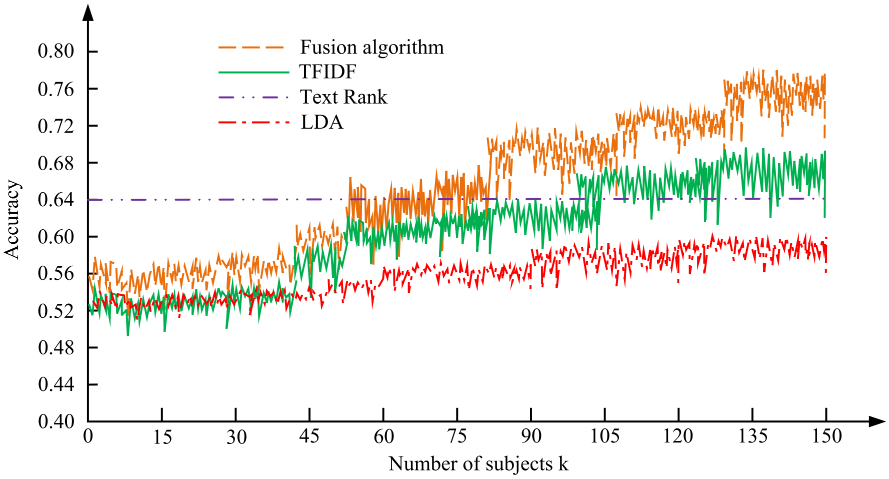
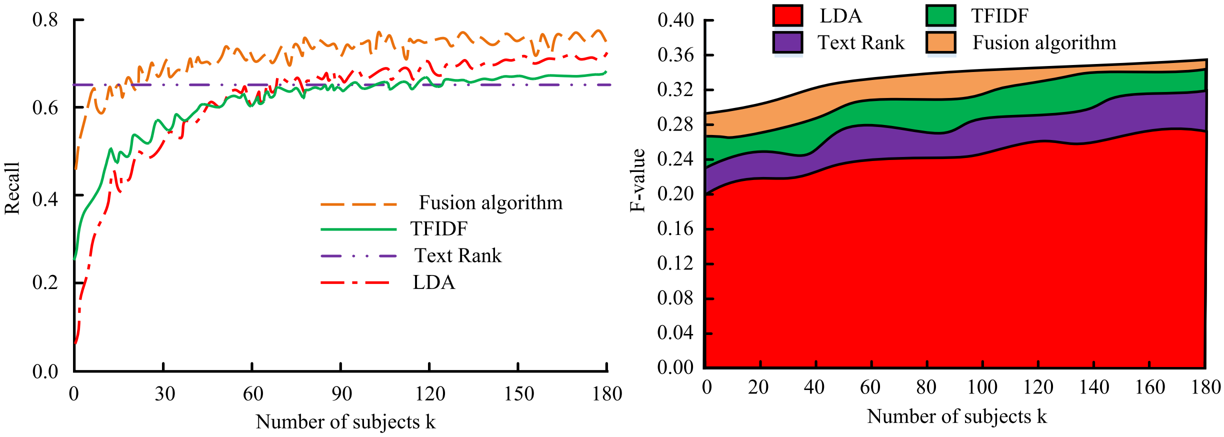


FIGURE 8. Accuracy of four algorithms under different k values.



(a) Recall rate of four algorithms

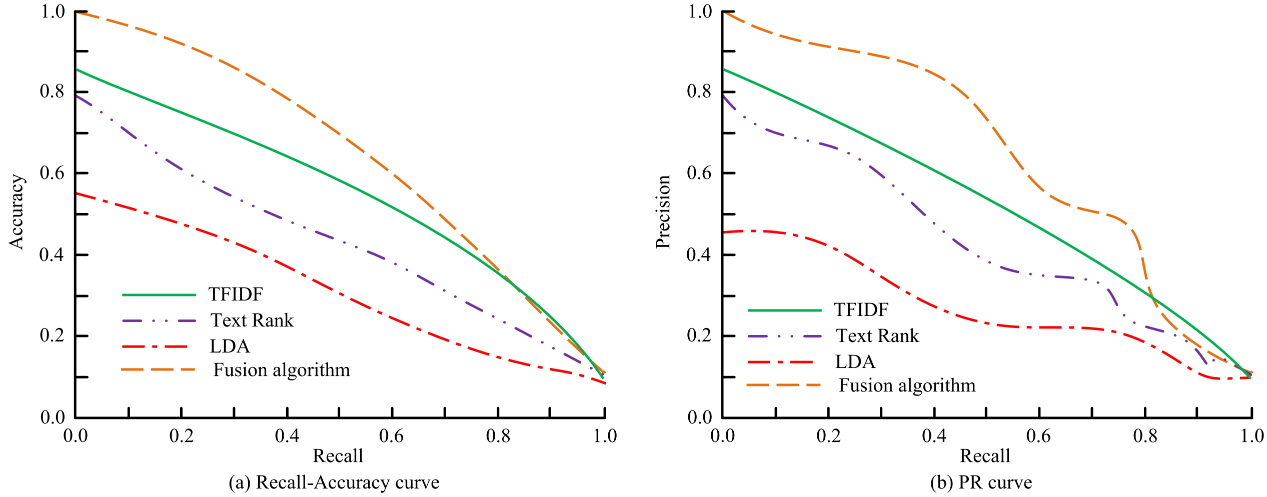
(b) F-value of four algorithms

FIGURE 9. Recall rate and F-value results of four algorithms.

are higher. Figure 10 shows the PR (Precision-Recall) curves and accuracy-recall curves of the four algorithms.

Figure 10(a) is the accuracy recall curve of the four algorithms. The accuracy recall curves of the four algorithms in Figure 10(a) show a downward trend, and the area under the curve of the fusion algorithm is much larger than that of the other three. Figure 10(b) shows the PR curves of the four algorithms. The overall PR curve of the four algorithms in Figure 10(b) also shows a downward trend, and the fusion algorithm has the largest area under the curve. In general, from the perspective of accuracy recall curve and PR curve dimensions, the performance of the fusion algorithm is superior to others. In line with the above dimensional comparison results, it is available that the proposed fusion algorithm has the best information extraction performance, and integrating this algorithm into IEM can effectively improve the extraction accuracy of IEM. Besides, the study also analyzed the word co-occurrence of the fusion algorithm and the extraction results of LDA and TextRank. The word co-occurrence between the extraction results of the three algorithms is Figure 11.

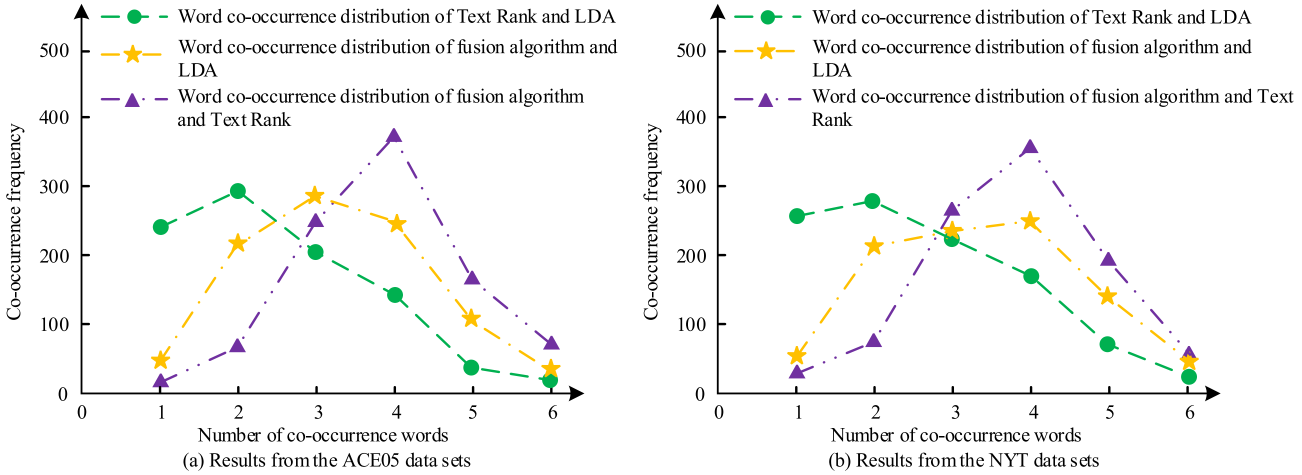
The co-occurrence showed in Figure 11 of the extraction results of the three algorithms in two datasets. Figure 11(a) shows the co occurrence of the three algorithms on the ACE05 dataset. In the comparison result between the fusion algorithm and the TextRank in Figure 11(a), the number of documents containing 4, 5, and 6 identical words in the six keywords is 386, 173, and 82, respectively. This result verifies that the fusion algorithm fully preserves the features of TextRank, which can fully utilize the structure of the document itself. Figure 11(b) shows the co-occurrence of the three algorithms in the NYT dataset. In the comparison result between the fusion algorithm and LDA in Figure 11(b), the documents' number containing 4, 5, and 6 identical words in the 6 keywords is 256, 162, and 63, respectively. The result demonstrates that the fusion algorithm also retains topic information to a certain extent. The above data shows that combining the advantages of TextRank and LDA, the



(a) Recall-Accuracy curve

(b) PR curve

FIGURE 10. PR curves and accuracy recall curves of four algorithms.



(a) Results from the ACE05 data sets

(b) Results from the NYT data sets

FIGURE 11. Co-occurrence among the extraction results of the three algorithms.

fusion algorithm has a better ability to extract information compared to the two basic algorithms. To verify the actual performance of IEMs incorporating fusion algorithms, the study tests IEMs on the ground of four different algorithms in the ACE05 dataset. Figure 12 shows the test results of this IEM.

As far as Figure 12 is concerned, the accuracy and precision of the IEM are 80.16% and 77.54%, respectively, which are significantly higher than the other three comparative models. This indicates that IEMs integrating TextRank and LDA have good extraction performance, which can be used to improve the current problem of IEM extraction accuracy. Based on the above comparative experiments, the number of datasets was increased to 600, and four models were compared again, with loss value, recall rate, F-value, error, and running time as evaluation indicators. The experimental results of loss values, recall rates and F-values for these four models are shown in Table 1.

According to Table 1, the minimum loss values for the LDA TextRank model, TFIDF model, Text Rank model, and LDA model are 0.071, 0.095, 0.161, and 0.209, respectively. The fusion algorithm information extraction model proposed in the study has the lowest loss value among the four models, with a value of 0.071, indicating the smallest difference between the information extracted by the model and the real information. The highest recall rates for the four models are 0.989, 0.931, 0.799, and 0.705, respectively. The LDA TextRank model has the highest recall rate among the four models, with a value of 0.989, which means that the correct number of information extraction accounts for the highest proportion of all actual samples. The maximum F-values of the four models are 0.949, 0.861, 0.791, and 0.686, respectively, with the LDA-TextRank model having the highest F-value of 0.942 among the four models. The experimental data show that the information extraction model proposed in this study outperforms the TFIDF model, Text Rank model,

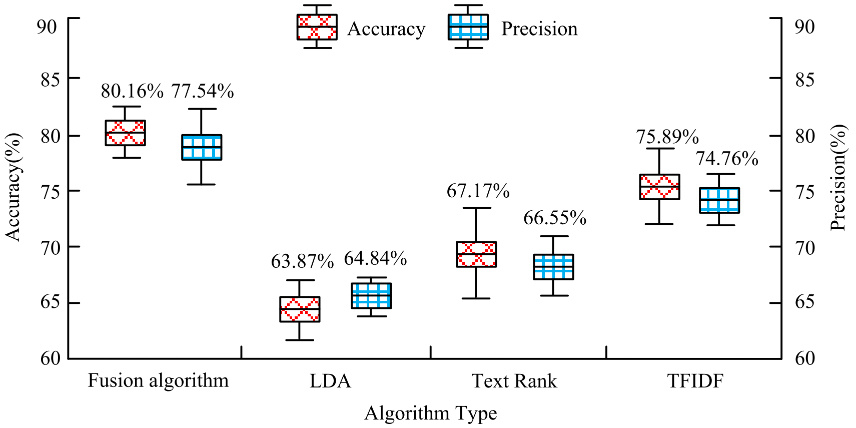


FIGURE 12. Actual test results of four extraction models.

TABLE 1. Comparison of the extraction performance of the four models.

Test number	Model	Magnitude of the loss	Recall	F value
The first experiment	Fusion algonrithm	0.074	0.989	0.942
	TFIDF	0.097	0.924	0.859
	Text Rank	0.165	0.799	0.765
	LDA	0.209	0.701	0.686
The second experiment	Fusion algonrithm	0.072	0.976	0.949
	TFIDF	0.099	0.931	0.834
	Text Rank	0.161	0.761	0.791
	LDA	0.211	0.705	0.657
The third experiment	Fusion algonrithm	0.71	0.971	0.947
	TFIDF	0.095	0.913	0.861
	Text Rank	0.164	0.786	0.790
	LDA	0.223	0.702	0.679
The fourth experiment	Fusion algonrithm	0.071	0.981	0.941
	TFIDF	0.095	0.921	0.859
	Text Rank	0.164	0.784	0.782
	LDA	0.221	0.701	0.667
The fifth experiment	Fusion algonrithm	0.075	0.975	0.939
	TFIDF	0.077	0.919	0.860
	Text Rank	0.167	0.779	0.779
	LDA	0.215	0.699	0.673

and LDA model in terms of loss value, recall rate, and F-value, indicating its superiority in information extraction. The running time and experimental errors of the LDA TextRank model, TFIDF model, Text Rank model, and LDA model in this experiment are shown in Figure 13.

As shown in Figure 13, the error rates of the LDA TextRank model, TFIDF model, Text Rank model, and LDA model are 23%, 32%, 38%, and 49%, respectively, and the running times are 210s, 465s, 426s, and 380s, respectively. The experimental results show that the model proposed by the research institute has the lowest error rate for information extraction, at 23%, and the model ran the fastest in the experiment, only 210 seconds, which is much faster than the other three models. The LDA TextRank model performs better than the other three models in terms of error and runtime, and has significant advantages in information extraction. Perform 500 iterations of the LDA TextRank model, the TFIDF model, the Text Rank model, and the LDA model according to the settings, and use the accuracy and loss values as evaluation indicators for this comparison experiment. The test results are shown in Figure 14.

As shown in Figure 14, the accuracy curve of the LDA TextRank model exhibits a fast convergence speed. After running the model for about 100 iterations, its accuracy has reached about 80% and then shows a slow upward trend. When the model is iterated about 200 times, the accuracy of the model begins to stabilize and finally stabilizes at 89%. The loss value curve of this model shows a characteristic of rapid decline at first, and then gradually stabilizes, and eventually the loss value of this model remains stable at 4.6%. The accuracy curve of the TFIDF model shows a slow increasing trend. After 300 iterations of the model, the accuracy reached 80% and remained stable. The model's accuracy ultimately reached 81%. The loss value curve of the TFIDF model also showed a slow downward trend, ultimately stabilizing at 7.3%. According to the accuracy curve of the Text Rank model, the accuracy of the model tends to stabilize after approximately 200 iterations. When the model was iterated around 300 times, there was a slow upward trend, and the final accuracy remained stable at 73%. The loss value curve of the Text Rank model shows a slow downward trend. From this curve, it can be seen that when the model runs around 200 iterations, a small range waveform appears in its loss value, and the final loss value stabilizes at 9.8%. According to the accuracy curve of the LDA model, the convergence speed of the model is fast but the accuracy is low. After running the model for about 50 iterations, the accuracy of the model reached about 50%, followed by a

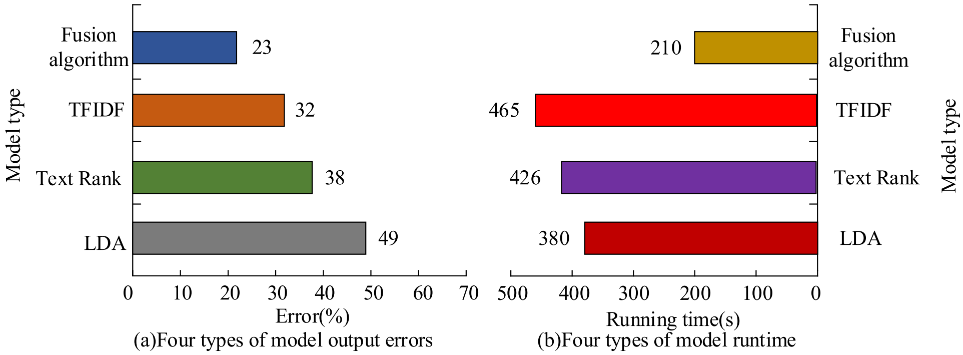


FIGURE 13. Error and running time.

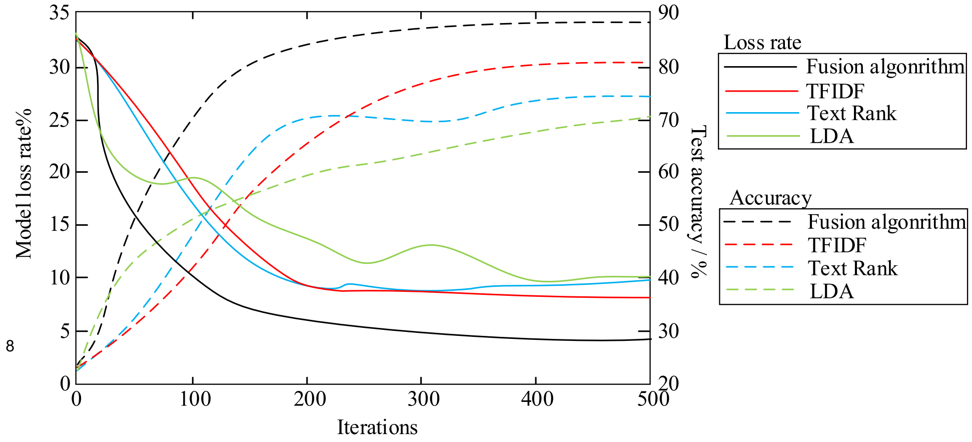


FIGURE 14. Accuracy and loss value curves.

slow increase in accuracy and finally stabilized at 71%. The overall loss value curve of this model shows a downward trend, with two large fluctuations, and the final loss value stabilizes at 10.1%. In summary, by comparing the accuracy and loss values of these four models after 500 iterations, the LDA TextRank model performs excellently in both aspects, which helps to improve the accuracy of information extraction.

V. CONCLUSION

Aiming at the problem that IEM is difficult to accurately extract key information nowadays, a hybrid TextRank algorithm and LDA topic model algorithm for IEM is raised. In this paper, a hybrid algorithm is obtained by fusing the LDA topic model and the TextRank algorithm. Then the hybrid algorithm is applied to IEM to improve its extraction accuracy. In the performance test of the fusion algorithm, it was found that the accuracy, recall, and F-value of the fusion algorithm were 76.1%, 77.0%, and 0.35, respectively, which were superior to the other three comparative algorithms. In addition, this article also found that the area under the PR curve of the fusion algorithm is larger than other algorithms, which leads to the conclusion that the performance of the fusion algorithm is much better than other comparative algorithms. In the empirical analysis of IEM, the accuracy rate of IEM is 80.16%, which is superior to the comparison model, and its precision is 77.54%, which is also superior to the comparison model. On the ground of the above experimental data, the IEM has a high accuracy rate of information extraction, so its application in content recognition, relationship extraction, automatic question answering and other fields will have good results. After increasing the test data samples, the loss value, recall rate and F value of this model are 0.071, 0.989 and 0.942, respectively, which are better than TFIDF model, Text Rank model and LAD model. In addition, in this experiment, the research proposed that the model has the shortest running time and the lowest error rate, respectively 210s and 23%, which is better than the other three models. After the iteration test, the accuracy rate and loss value of the LDA-TextRank model were stable at 89% and 4.6%, respectively, which were better than the other three models used for comparison. The above results show that the information extraction model proposed in this study has a high accuracy of information extraction, so its application in content recognition, relationship extraction, automatic question answering and other fields will have a good effect. The algorithm of this research still has the shortcoming of not being able to accurately express the main idea meaning of documents. How to use deep learning algorithm to learn the main idea meaning of documents is the future research direction.

REFERENCES

[1] L. Li, "Abnormal detection method of accounting data based on information extraction technology," *Int. J. Inf. Commun. Technol.*, vol. 21, no. 1, pp. 63–78, 2022.

[2] Y. G. Vasin and D. Y. Vasin, "An intelligent information technology for symbol-extraction from weakly formalized graphic documents," *Pattern Recognit. Image Anal.*, vol. 29, no. 1, pp. 51–57, Jan. 2019.

[3] X. Zhang, L. Ma, and K. Peng, "A novel key performance indicator oriented process monitoring method based on multiple information extraction and support vector data description," *Can. J. Chem. Eng.*, vol. 100, no. 5, pp. 1013–1025, May 2022.

[4] P. Zhu, J. Hu, X. Li, and Q. Zhu, "Using blockchain technology to enhance the traceability of original achievements," *IEEE Trans. Eng. Manag.*, vol. 70, no. 5, pp. 1693–1707, May 2023.

[5] G. Tür, D. Hakkani-Tüer, and K. Oflazer, "A statistical information extraction system for Turkish," *Natural Lang. Eng.*, vol. 9, no. 2, pp. 181–210, Jun. 2003.

[6] A. Culotta, T. Kristjansson, A. McCallum, and P. Viola, "Corrective feedback and persistent learning for information extraction," *Artif. Intell.*, vol. 170, nos. 14–15, pp. 1101–1122, Oct. 2006.

[7] X. Yang, J. Na, G. Tang, T. Wang, and A. Zhu, "Bank gully extraction from DEMs utilizing the geomorphologic features of a Loess hilly area in China," *Frontiers Earth Sci.*, vol. 13, no. 1, pp. 151–168, Mar. 2019.

[8] R. Cai, Z. Lin, W. Chen, and Z. Hao, "Shared state space model for background information extraction and time series prediction," *Neurocomputing*, vol. 468, pp. 85–96, Jan. 2022.

[9] I. Ferjani, M. S. Hidri, and A. Frihida, "SiNoptiC: Swarm intelligence optimisation of convolutional neural network architectures for text classification," *Int. J. Comput. Appl. Technol.*, vol. 68, no. 1, pp. 82–100, 2022.

[10] 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, Apr. 2022.

[11] T. Karthikeyan, K. Sekaran, D. Ranjith, K. Vinoth, and J. M. Balajee, "Personalized content extraction and text classification using effective web scraping techniques," *Int. J. Web Portals*, vol. 11, no. 2, pp. 41–52, Jul. 2019.

[12] S. Zhang, Q. Luo, Y. Feng, K. Ding, D. Gifu, S. Zhang, X. Ma, and J. Xia, "Keyphrase 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, Dec. 2021.

[13] M. F. Fakhrezi, M. A. Bijaksana, and A. F. Huda, "Implementation of automatic text summarization with TextRank method in the development of al-Qur'an vocabulary encyclopedia," *Proc. Comput. Sci.*, vol. 179, pp. 391–398, 2021.

[14] M. Bordoloi, P. C. Chatterjee, S. K. Biswas, and B. Purkayastha, "Keyword extraction using supervised cumulative TextRank," *Multimedia Tools Appl.*, vol. 79, nos. 41–42, pp. 31467–31496, Nov. 2020.

[15] J. Li, G. Huang, C. Fan, Z. Sun, and H. Zhu, "Key word extraction for short text via word2vec, doc2vec, and textrank," *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 27, no. 3, pp. 1794–1805, May 2019.

[16] N. Zhou, R. D. Brook, I. D. Dinov, and L. Wang, "Optimal dynamic treatment regime estimation using information extraction from unstructured clinical text," *Biometrical J.*, vol. 64, no. 4, pp. 805–817, Apr. 2022.

[17] J. L. Martinez-Rodriguez, I. Lopez-Arevalo, and A. B. Rios-Alvarado, "Mining information from sentences through semantic web data and information extraction tasks," *J. Inf. Sci.*, vol. 48, no. 1, pp. 3–20, Feb. 2022.

[18] S. Sun and J. Ren, "GASF–MSNN: A new fault diagnosis model for spatiotemporal information extraction," *Ind. Eng. Chem. Res.*, vol. 60, no. 17, pp. 6235–6248, May 2021.

[19] H. Liu, X. Yuan, and Y. J. Zhang, "Statistical beamforming for FDD downlink massive MIMO via spatial information extraction and beam selection," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4617–4631, Jul. 2020.

[20] J. M. Steinkamp, C. Chambers, D. Lalevic, H. M. Zafar, and T. S. Cook, "Toward complete structured information extraction from radiology reports using machine learning," *J. Digit. Imag.*, vol. 32, no. 4, pp. 554–564, Aug. 2019.

[21] X. Fei, "An LDA based model for semantic annotation of web English educational resources," *J. Intell. Fuzzy Syst.*, vol. 40, no. 2, pp. 3445–3454, Feb. 2021.

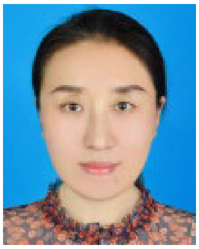
[22] Z. Yang, H. Yu, J. Tang, and H. Liu, "Toward keyword extraction in constrained information retrieval in vehicle social network," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4285–4294, May 2019.

[23] X. Wang, B. Sun, and H. Dong, "Domain-invariant adversarial learning with conditional distribution alignment for unsupervised domain adaptation," *IET Comput. Vis.*, vol. 14, no. 8, pp. 642–649, 2020.

[24] C. Gulden, M. Kirchner, C. Schüttler, M. Hinderer, M. Kampf, H.-U. Prokosch, and D. Toddenroth, "Extractive summarization of clinical trial descriptions," *Int. J. Med. Informat.*, vol. 129, pp. 114–121, Sep. 2019.

[25] R. Grishman, "Information extraction," *IEEE Intell. Syst.*, vol. 30, no. 5, pp. 8–15, Sep. 2015.

[26] P. Zhu, J. Hu, Y. Zhang, and X. Li, "Enhancing traceability of infectious diseases: A blockchain-based approach," *Inf. Process. Manage.*, vol. 58, no. 4, Jul. 2021, Art. no. 102570.



YUNBO WEI was born in Heilongjiang, China, in 1979. She received the B.S. degree in mathematics and applied mathematics from Heilongjiang University, Harbin, China, in 2003, and the M.S. degree in applied mathematics from Jilin University, Changchun, China, in 2009.

From 2003 to 2008, she was a Teaching Assistant with the Mathematical Department, Qiqihar University, Heilongjiang. Since 2009, she has been a Lecturer with the Department of Information and Computing Science, Qiqihar University. She is the author of one book and more than five articles. Her research interests include mathematical modeling, statistical optimization, data mining, optimization algorithm, and information theory.

YONGSHENG DING was born in Heilongjiang, China, in 1974. He received the B.S. degree in applied mathematics from Qiqihar University, Heilongjiang, in 1998, and the M.S. degree in computational mathematics from Northwestern Polytechnical University, Shanxi, China, in 2005. From 2005 to 2009, he was a Lecturer with the Mathematical Department, Qiqihar University. Since 2010, he has been an Associate Professor with the Department of Mathematics and Information Science, Qiqihar University. He is the author of two books and more than 20 articles. His research interests include mathematical modeling, digital image processing, and computer graphics.