applied sciences

MDPI

Article

IWF-TextRank Keyword Extraction Algorithm Modelling

Liyan Zhang [1], Wenhui Wang [2], Jian Ma [1,*] and Yuan Wen [1]

[1] School of Civil Engineering, Suzhou University of Science and Technology, Suzhou 215011, China; outerspace@mail.usts.edu.cn (L.Z.); 18336284660@163.com (Y.W.)
[2] School of Business, Suzhou University of Science and Technology, Suzhou 215009, China; 13032531656@163.com
* Correspondence: 2216@mail.usts.edu.cn

Abstract: Keywords are used to provide a concise summary of the text, enabling the quick understanding of core information and assisting in filtering out irrelevant content. In this paper, an improved TextRank keyword extraction algorithm based on word vectors and multi-feature weighting (IWF-TextRank) is proposed to improve the accuracy of keyword extraction by comprehensively considering multiple features of words. The key innovation is demonstrated through the application of a backpropagation neural network, combined with sequential relationship analysis, to calculate the comprehensive weight of words. Additionally, word vectors trained using Word2Vec are utilised to enhance the model's semantic understanding. Finally, the effectiveness of the algorithm is verified from various aspects using traffic accident causation data. The results show that this algorithm demonstrates a significant optimisation effect in keyword extraction. Compared with the traditional model, the IWF-TextRank algorithm shows significant improvement in accuracy (*p*-value), recall (R-value), and F-value.

Keywords: TextRank; keyword extraction; word vectors; multivariate feature weighting

Citation: Zhang, L.; Wang, W.; Ma, J.; Wen, Y. IWF-TextRank Keyword Extraction Algorithm Modelling. *Appl. Sci.* **2024**, *14*, 10657. https://doi.org/10.3390/app142210657

Academic Editor: Douglas O'Shaughnessy

Received: 21 October 2024
Revised: 11 November 2024
Accepted: 13 November 2024
Published: 18 November 2024

1. Introduction

In the digital age, the rapid proliferation of data presents challenges in efficiently identifying valuable information from large volumes of irrelevant content. This problem is particularly acute in specialised fields such as traffic accident cause analysis, where relevant keywords must be accurately extracted to summarise extensive textual datasets. Keyword extraction techniques constitute a fundamental tool in natural language processing (NLP) and play a critical role in addressing these challenges by assisting users in promptly identifying core content [1]. Currently, keyword extraction methods can generally be classified into two categories: supervised and unsupervised keyword extraction [2,3].

Supervised keyword extraction methods typically approach the process as a binary classification task, extracting keywords by training models on manually labelled corpora [4,5]. Typically, complex machine learning algorithms are employed to learn from rich feature sets and optimise keyword prediction performance. However, these approaches are constrained by their requirement for substantial annotated datasets, which are often difficult to acquire [6]. Furthermore, models trained on specific datasets may not generalise well to various document types or keyword extraction tasks, resulting in potential overfitting. In contrast, unsupervised methods do not require labelled data; instead, they rank keywords based on specific metrics, allowing unsupervised methods to be more adaptable to different contexts. In response to the limitations of traditional unsupervised approaches, Mihalcea and Tarau [7] proposed the TextRank algorithm in 2004, which has since gained widespread attention. TextRank is a graph-based algorithm that constructs co-occurrence networks by treating words as nodes and their co-occurrence relations as edges. However, the algorithm assumes that all words are equally important initially, which might not fully capture the semantic and syntactic significance of individual words within a document. To overcome

these constraints, this paper proposes an improved method, the IWF-TextRank algorithm, which integrates a variety of lexical features (e.g., semantic vectors, word frequency, lexical properties, and word length) into the traditional TextRank framework. The innovation of this method lies in the dynamic allocation of word weights through backpropagation (BP) neural networks and sequence relationship analysis. This method can capture more nuanced contextual information, thereby improving the relevance and accuracy of keyword extraction, especially on domain-specific datasets such as traffic accident reports.

The main contribution of this article is to propose a new keyword extraction algorithm, IWF-TextRank, which extends the traditional TextRank into a multi-feature weighted framework. It integrates features such as semantic vectors, word frequency, lexicality, and word length, and combines them with a BP neural network to achieve dynamic weight distribution, which significantly improves the keyword extraction effect of the algorithm in data in specific fields (such as traffic accident reports). The specific features are as follows:

- Semantic vectors: word vectors are utilised to take into account the semantic relationships between words;
- Word frequency and lexicality: adjusts the importance of keywords based on their frequency of occurrence and grammatical roles in the sentence;
- Word length: word length is taken into account as longer words usually carry more information;
- Backpropagation neural network: a BP neural network is used to dynamically adjust word weights to ensure that more important, contextually relevant words are emphasised.

In addition, by adding sequence relationship analysis to the BP neural network, IWF-TextRank can more accurately capture the contextual association of the text, ultimately achieving higher precision and recall rates. This improved approach aims to provide more accurate domain-specific keyword extraction, addressing the shortcomings of traditional algorithms such as TF-IDF and basic TextRank, which cannot take into account semantic relationships and contextual importance. Experimental results show that the IWF-TextRank algorithm outperforms traditional methods such as TF-IDF and basic TextRank in terms of keyword extraction accuracy, providing an effective solution for keyword extraction tasks in fields such as traffic analysis.

Keyword extraction methods are mainly classified into three categories: statistical-feature-based methods, topic-based modelling methods, and word graph modelling-based methods. Among the statistical-based methods, TF-IDF [8] is a widely used technique where keywords are ranked based on word frequency and inverse document frequency. However, TF-IDF has been criticised for its over-reliance on word frequency, especially in professional domains, which often leads to poor extraction results [9]. To address this issue, researchers have proposed various improvements, such as integrating positional and word span weights [10], or combining TF-IDF with weight-balancing algorithms [11]. Topic-based modelling approaches, such as Latent Dirichlet Allocation (LDA [12]), have also been used for keyword extraction. LDA performs well in capturing semantic associations between words, making it a powerful tool for extracting topic-relevant keywords from large text corpora [13,14]. However, LDA is computationally expensive and may not perform well when dealing with short texts or single topic documents [15,16]. Finally, one of the most widely used word-graph-based keyword extraction methods is TextRank [7]. This algorithm simulates the PageRank algorithm through a co-occurrence network, iteratively calculates the importance scores of the nodes, and extracts the words with higher scores as keywords. In recent years, researchers have proposed a number of extensions to TextRank, such as SW-TextRank and DK-TextRank, which incorporate semantic features and optimise word weights, but these methods often fail to fully balance multiple linguistic features [17,18].

In recent years, researchers have proposed a variety of improved algorithms to improve the accuracy and adaptability of keyword extraction. Among the existing methods, NE-Rank and SemanticRank are typical representatives based on semantics and word weight optimisation. NE-Rank mainly enhances the importance score of words by word

frequency, thereby improving the accuracy of keyword extraction. However, due to the lack of consideration of the semantic relationship between words, NE-Rank performs poorly when processing complex texts or data in specific fields [19]. SemanticRank introduces word vector technology to improve the accuracy of keyword extraction through semantic similarity [20]. However, its performance in specific domain data is limited because the semantic modelling of this method tends to be of general context [21]. In contrast, the IWF-TextRank proposed in this paper comprehensively considers multiple features such as word frequency, lexicality, word length, and semantics, and combines the BP neural network to optimise weights, which is more suitable for keyword extraction in specific fields. Table 1 compares the main features and performance of NE-Rank, SemanticRank, and IWF-TextRank to more clearly show the innovations and advantages of IWF-TextRank.

Table 1. Keyword extraction algorithm performance comparison table.

Algorithm	Semantic Integration	Weighting Method	Performance
NE-Rank	no	Simple weighting based on word frequency	Unable to capture deep semantic relationships
SemanticRank	yes	Context-based semantic weighting	Applicable to general-context semantics
IWF-TextRank	yes	Multi-factor weights and BP neural network	Excellent extraction effect, especially suitable for data in specific fields

The remainder of this paper is structured as follows: Section 2 describes the IWF-TextRank modelling framework and the methodology used in this study. Section 3 discusses the experimental results and analyses them in comparison with existing methods. Finally, Section 4 summarises the contributions of this paper and suggests potential directions for future research.

2. Materials and Methods

2.1. Classical TextRank Algorithm

The TextRank algorithm is improved on the basis of the PageRank [22] algorithm. The algorithm takes words as nodes and the co-occurrence relationship between words as edges to construct a network, and then calculates the TextRank value of words and ranks them [23]. The higher the TextRank value, the higher the importance of the word, the higher the probability of becoming a keyword. Assuming the existence of word i, the score of word i calculated by the TextRank algorithm is:

$$WS(V_i) = (1-d) + d \times \sum_{V_j \in In(V_i)} \frac{W_{ji}}{\sum V_k \in Out(V_j) W_{jk}} WS(V_j) \quad (1)$$

where $In(V_i)$ represents the set of other words pointing to word i; $Out(V_j)$ is the set of other words pointing to word i; $WS(V_i)$ represents the weight of word i in the last iteration; $WS(V_i)$ denotes the weight of word i in the last iteration; the similarity of words i and j is denoted by W_{ji}; and $WS(V_j)$ denotes the weight of word j in the last iteration.

2.2. Overview of the IWF-TextRank Algorithm Model

The classic TextRank algorithm is processed using the text as a unit and does not require prior dataset training. The principle is simple and easy to operate, but there are obvious shortcomings:

- It is influenced by high-frequency words, and therefore these need to be screened in combination with other features in order to achieve a better result;
- The initial weight of each word node is set to 1 by default, and, according to the actual situation of each word node, the weights are not the same [24];
- Word weight evaluation based on co-occurrence relations may not fully capture the deep semantic associations between words [25].

In order to overcome the shortcomings of the classical TextRank algorithm, this paper proposes an improved TextRank keyword extraction algorithm based on word vectors and multi-feature weighting (IWF-TextRank), which integrally considers the multiple features of words to improve the accuracy of keyword extraction. The algorithm flow is shown in Figure 1. The IWF-TextRank algorithm mainly consists of four steps:

1. Text preprocessing. The raw traffic accident data undergo segmentation, duplication removal, lexical tagging, and filtering to form the initial candidate keyword set. Jieba, a popular Chinese text segmentation tool, is used with a custom dictionary to ensure the accurate recognition of domain-specific vocabulary.
2. Multivariate feature extraction. Statistical computational work is performed on each feature to determine the level of feature importance.
3. Comprehensive weight calculation. On the basis of determining the importance of features, the comprehensive weight of words is calculated by combining the BP neural network with the ordinal relationship analysis method.
4. Construction of word map and extraction of keywords. The calculated word weights are used as inputs to the TextRank algorithm and, based on this, the word map model is constructed using the word co-occurrence relationship, and then the keywords are extracted.

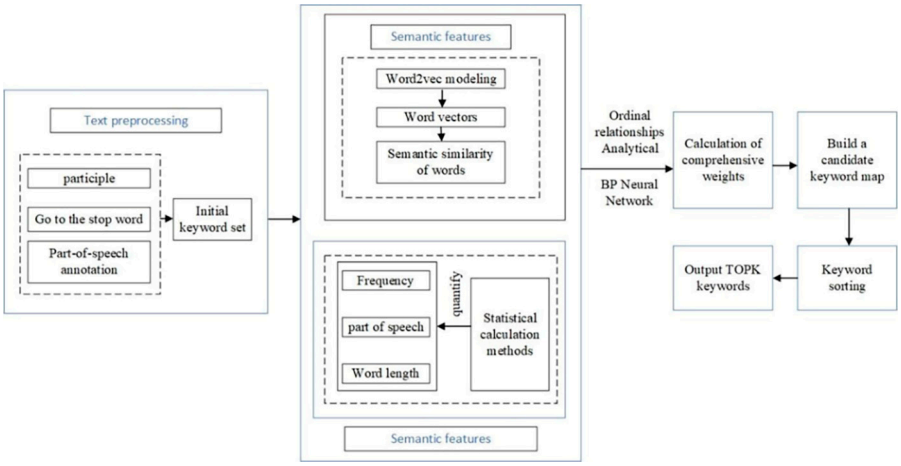


Figure 1. Flowchart of the IWF-TextRank algorithm.

2.3. Establishment of the IWF-TextRank Model

The TextRank algorithm is a word graph model constructed based on text features. The algorithm considers the words obtained after text preprocessing as nodes and then constructs a candidate keyword graph, G = (V, E), where V is the set of nodes of the candidate keywords and E is the set of edges of the candidate keywords. Whether there is an edge between two keywords depends on whether there is a co-occurrence relationship between the two. There is only an edge between two nodes if their corresponding words co-occur in a window of size K. After the word graph is constructed, iteration is required to calculate the importance of the words until convergence and to sort them and output the Top K Ranked (TOPK) keywords as the keywords of the text. In order to improve the semantic capture ability of the TextRank algorithm, introducing the Continuous Bag of Words (CBOW) model is an effective strategy. While traditional TextRank focuses

on statistical features such as word co-occurrence, the CBOW model generates vector representations for words through the Word2Vec framework, thus revealing the deep semantic connections between words. By combining the output of the ordinal relationship analysis method and the BP neural network, the comprehensive weight of each word can be obtained. On the basis of the classical TextRank algorithm, the combined weights of the calculated words are used as inputs, and the IWF-TextRank algorithm formula can be obtained as:

$$G(v_i) = (1-d)w(v_i) + d \times w(v_i) \sum_{v_j \in Out(v_i)} \frac{w_{ij}}{\sum_{v_i \in Out(v_j)} w_{jk}} G(v_j) \quad (2)$$

where $w(v_i)$ represents the combined weight of the words.

2.4. Feature Analysis Model

In this section, statistical feature modelling and semantic feature modelling in the IWF-TextRank algorithm are discussed. The following subsections describe the specific analysis methods for different features, respectively.

2.4.1. Statistical Feature Modelling

In the IWF-TextRank algorithm, statistical feature modelling is used to quantify the importance of words in the text, aiming to comprehensively evaluate the keyword potential of words from multiple statistical perspectives. Specifically, statistical feature modelling takes into account word frequency, lexical features, and word length. These features help construct the basic importance score of each word and provide data support for the subsequent comprehensive weight calculation.

(1) Word frequency characteristics

The term frequency feature uses the classic TF-IDF (term frequency–inverse document frequency) method to measure the relative importance of words in the entire text dataset. TF-IDF is mainly composed of term frequency (TF) and inverse document frequency (IDF) [26]. The term frequency (TF), which represents the frequency of occurrence of the word t_i in the document D_j, is calculated as:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (3)$$

where $n_{i,j}$ refers to the number of times the word appears in document D_j; $\sum_k n_{k,j}$ represents the sum of the occurrences of all words in the document D_j.

The inverse document frequency (IDF) measures the importance of a word in the whole corpus and is obtained by dividing the total number of documents in the database by the total number of documents containing the word and taking the logarithm of the base 10. The formula is:

$$idf_i = log \frac{|\mathrm{D}|}{|\{j : t_i \in d_j\}|} \quad (4)$$

where $|\mathrm{D}|$ denotes the total number of documents in the corpus, and $\sum_k n_{k,j}$ denotes the total number of documents in the corpus containing the word t_i.

Multiplying TF and IDF gives the TF-IDF value of each word in the document, and then assigns preliminary weights to the words in the keyword candidate set in IWF-TextRank.

$$TF - IDF = tf_{i,j} \times idf_i \quad (5)$$

(2) Lexical features (posi)

Lexical features are used to identify the grammatical properties of words, thereby distinguishing key categories such as nouns, verbs, and adjectives, which usually have different importance in text analysis. For example, nouns and verbs are usually more representative when describing events, so they are given higher weights. In the statistical

process, we determine the importance of each type of part of speech through part-of-speech statistical methods and adjust the priority of different parts of speech based on the statistical results.

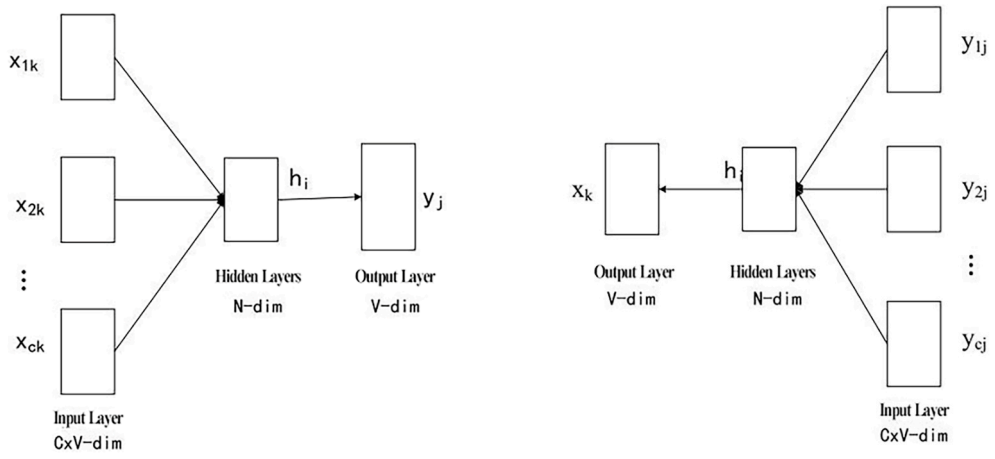
(3) Word length features (leni)

Word length is another statistical feature used to evaluate the amount of information a word contains in a text. Generally, longer words tend to carry more information and may be more suitable as keywords. In word length feature modelling, the word length is divided into five levels (e.g., 1, 2, 3, 4, >4), and corresponding weights are assigned to different word lengths based on statistical results, thereby ensuring that word length features have an appropriate impact on keyword extraction.

2.4.2. Semantic Feature Modelling

(1) Word2vec algorithm

Word2vec is a natural language processing model proposed by Mikolov [27] in 2013, which contains two training models, Skip-gram and Continuous Bag of Words (CBOW) [28]. This paper selects the CBOW model in Word2Vec to vectorise semantics. As shown in Figure 2a, the core idea of the CBOW model is to predict the probability distribution of the target word through the words in the context. It is essentially a three-layer neural network, including an input layer, a hidden layer, and an output layer.



(**a**) (**b**)

Figure 2. Working diagram of CBOW model and Skip-gram model. (**a**) CBOW model; (**b**) Skip-gram model.

1. Input layer: The input layer converts known context words into a computer-recognisable form, that is, into a C × V one-hot tensor X. Here, C represents the number of context words and V is the size of the dictionary. One-hot encoding is only used to solve the problem that text words cannot be directly processed by computers, and there is no need to represent the relationship between words.
2. Hidden layer: The role of the hidden layer is to reduce the dimensionality of the input word vector. By performing matrix multiplication with a V × N weight matrix, the C × V one-hot tensor is converted to a C × N vector representation, where N is the length of the word vector. Then, the C word vectors are added together to obtain a 1 × N vector to represent the entire context. This dimensionality reduction process retains semantic information while reducing computational complexity.
3. Output layer: The output layer contains a softmax classifier, which multiplies the 1 × N vector obtained from the hidden layer with the N × V output weight matrix to obtain a 1 × V vector, which represents the predicted probability distribution of the target word. The vector is normalised by the softmax function to obtain the probability distribution of the target word.

In contrast, as in Figure 2b, the Skip-gram model reverses the use of known target words to predict their contexts, which is suitable for large corpora. The CBOW model is suitable for corpora with high domain specialisation due to its higher computational accuracy and shorter time consumption. In this study, when processing the text of traffic accident data, the CBOW model achieves higher computational efficiency in a shorter time while being able to accurately generate the semantic vectors of the text. These word vectors are used for subsequent semantic feature computation to enhance keyword extraction.

(2) Semantic features of words

The classical keyword extraction algorithms ignore the semantic information between words, while the context of words and the articulation between sentences and between words can be reflected by semantic information. Therefore, this paper incorporates the semantic features of words based on the consideration of word frequency, word nature, and word length features, and uses the CBOW model in Word2vec to train the data obtained from the dataset to obtain the word vectors, and combines with the cosine calculation formula to calculate the similarity of the words. Assuming the existence of two words, I and J, the similarity between words is calculated as:

$$Sim(I,J) = cos\theta = \frac{I.J}{|I||J|} = \frac{\sum_{I=1}^{n} I_i.J_i}{\sqrt{\sum_{i=1}^{n}(I_i)^2} \times \sqrt{\sum_{i=1}^{n}(J_i)^2}} \quad (6)$$

where I_i represents the components of vector I, and J_i represents the components of the vector J.

The semantic weight of a word can be further calculated after obtaining the word similarity. The formula is as follows:

$$w_{sim}(v_i) = \sum_{v_j \in V} \frac{Sim(v_i, v_j)}{|V|} \quad (7)$$

where $w_{sim}(v_i)$ represents the semantic weight of node i; $Sim(v_i, v_j)$ represents the semantic weight of semantic similarity between node i and node j; and V represents the set of nodes.

2.5. Combined Word Weight Model

In order to more accurately determine the comprehensive weight of words in the keyword extraction process of the IWF-TextRank algorithm, this section proposes a multi-level weighting method that combines sequential relationship analysis and BP neural network optimisation. Specifically, Section 2.5.1 introduces the sequential relationship analysis method to determine the relative importance of different indicators and calculate preliminary weights; Section 2.5.2 explains, in detail, the application of the BP neural network to optimise the weight distribution of word features through training to improve the accuracy of the model; and Section 2.5.3 combines the above methods to calculate the final comprehensive word weight to ensure that the keyword extraction process can accurately capture the semantics and context of the text.

2.5.1. Sequential Relationship Analysis

In determining the weights of the indicators by using the method of ordinal relationship analysis (G_1 assignment method) [29], it is necessary for experts in the relevant fields to determine and rank the importance of each indicator based on their experience and the characteristics of the research object, and then assign a value to the indicators based on their degree of importance to calculate the weights of each indicator. The specific process is as follows:

- Defining the indicator importance ranking. Assume that there are n evaluations, whose importance is expressed by M. If the importance of the evaluation indicator M_i

is greater than that of M_j, i.e., $M_i > M_j$, the importance of the n evaluation indicators can be sequentially ranked as:

$$M_1 > M_2 > M_3 > \cdots > M_k > \cdots > M_n \tag{8}$$

- Determining the ratio of importance. After obtaining the ranking of the importance of indicators, experts determine the importance ratio between neighbouring indicators according to experience r_k. The regularity of its value is shown in Table 2.

$$r_k = \frac{M_{k-1}}{M_k} \tag{9}$$

where k = n, n − 1, n − 2,. . ., 3, 2.

- Calculating the weighting factor. After determining the value of the indicator, the weight of the indicator can be calculated according to the following formula:

$$W_n = \left(1 + \sum_{k=2}^{n} \prod_{i=k}^{n} r_i\right)^{-1} \tag{10}$$

Table 2. Table of values for the ratio of importance between indicators.

r_k	Instructions
1.0	Indicators M_{k-1} and M_k are equally important.
1.1	Indicators M_{k-1} and M_k are equally and marginally important.
1.2	Indicator M_{k-1} Slightly more important than M_k
1.3	Indicators M_{k-1} and M_k are both between marginally and significantly important.
1.4	Indicators M_{k-1} and M_k are clearly important.
1.5	Indicators M_{k-1} and M_k are both between clearly and strongly important.
1.6	Indicators M_{k-1} and M_k are strongly important.

2.5.2. BP Neural Network

In the selection of weighting coefficients, this paper optimises the weights of each feature through a BP neural network [30] (backpropagation algorithm) to avoid the subjectivity of manually setting weights. The BP neural network is a kind of error direction propagation multilayer feedback neural network, which is composed of three parts: an input layer, hidden layer, and output layer. Each layer is connected by neurons, and the core idea is to train the neural network with gradient descent function and then realise the weight update of the inter-layer network, which is divided into the forward-error-seeking and reverse-error-propagation processes.

1. Forward error calculation

Assume that the input layer data are X = {X_1, X_2, X_3,. . ., X_n}, the output layer data are Y = O = {O_1, O_2, O_3,. . ., O_n}, and the hidden layer output data are Y = { Y_1, Y_2, Y_3,. . ., Y_n}; the expected output data are H = {H_1, H_2, H_3,. . ., H_n}, the initial weight between the input layer and the hidden layer is w_1 and the bias term is a_1, and the initial weight between the hidden layer and the output layer is w_2 and the bias term is a_2. The data X = {X_1, X_2, X_3,. . ., X_n} are input into the neural network from the input layer, and then weighted by w_1 and a_1 and input into the hidden layer. After being processed by the activation function, the output is Y = {Y_1, Y_2, Y_3,. . ., Y_n}; then, Y is weighted by w_2 and a_2 and transmitted to the output layer. After being processed by the activation function, the output data O = {O_1, O_2, O_3,. . ., O_n} are obtained. The error E between O = {O_1, O_2, O_3,. . ., O_n} and the expected output value H = {H_1, H_2, H_3, . . ., H_n} is calculated.

2. Error backpropagation

After the error E is calculated by forward propagation, if the error does not reach the specified range, the network will propagate the error from the output layer to the input layer through the hidden layer. During the backpropagation process, the network will

distribute the error to each layer of neural units, calculate the error gradient of each layer via the gradient descent method, and adjust the weight parameters of each layer according to the error gradient. After multiple iterations, the network weights are continuously optimised until the error E reaches the preset range, indicating that the network training is complete. At this point, the trained network can calculate new data.

The BP neural network can accurately fit nonlinear data without knowing the relationship between the input and output data in advance. The network automatically discovers the potential rules between the input and output data through strong learning ability, and adjusts the weights accordingly. In addition, the BP neural network has strong fault tolerance, and even if there are errors in the data, the training effect will not be significantly affected. However, the training process for the BP neural network requires a large amount of data to ensure the stability and reliability of the network weights. This paper takes a large amount of traffic accident data as the research object, and considering the advantages of BP neural network, these data are selected to calculate the weight parameters between words.

2.5.3. Combined Word Weights

The weight of each feature of a word needs to be calculated before the comprehensive weight calculation, and this paper uses the sequential relationship analysis method to determine the weight of word features. According to the accidental text features and the experience of experts and scholars, it is determined that the order of importance of word-level features is as follows: TFIDF > Lexicality > Word length > Semantics, which are recorded as A, B, C, D, respectively, and then the weights of the four indicators are recorded as W_A, W_B, W_C, W_D, and the values of the importance degree r_k among the indicators are shown in Table 3.

$$\begin{aligned} W_D &= (1 + 1.2 \times 1.4 \times 1.2 + 1.4 \times 1.2 + 1.2)^{-1} = 0.169 \\ W_C &= 0.169 \times 1.2 = 0.204 \\ W_B &= 0.204 \times 1.4 = 0.285 \\ W_A &= 0.285 \times 1.2 = 0.342 \end{aligned} \tag{11}$$

Table 3. Table of values for r_k.

r_k	Retrieved Value
r_2	1.2
r_3	1.4
r_4	1.2

After determining the importance weights of the first-level features, the second-level features under the word and word length can be determined according to the statistical results of the word's lexical and word length. The second-level features are recorded as A_i, B_i, C_i, D_i, where A_i is the TFIDF value of the word, and the D_i word is the w_{sim} value. Then, the word frequency, word length, and semantic weight of the words can be obtained as:

$$\begin{aligned} W_{freq} &= 0.342 \times TFIDF_i \\ W_{pos} &= 0.285 \times B_i \\ W_{len} &= 0.204 \times C_i \\ W_{sem} &= 0.169 \times W_{sim} \end{aligned} \tag{12}$$

The weights of the word features obtained using the sequential relationship analysis method are more subjective and the accuracy rate is relatively low; thus, in this paper, we further set parameters to weight the features on the basis of the four features, which are noted as α, β, γ, and δ, and use the BP neural network to optimise the weight allocation of the four features.

When training with BP neural networks, the word weights obtained from the above calculations are combined to provide the input of the neural network and determine whether it is a keyword as the output, where 1 means it is a keyword and 0 means it is not a keyword. The pre-processed text is iteratively calculated by the BP neural network until convergence, after which the feature parameters are obtained and used in the calculation of the integrated weights with the formula:

$$W = \alpha W_{freq} + \beta W_{pos} + \gamma W_{len} + \delta W_{wem} \tag{13}$$

where α refers to the word frequency parameter; β refers to the lexicality parameter; γ refers to the word length parameter; δ refers to the semantic parameter; W_{freq} indicates the word frequency weights of words; W_{pos} indicates the lexical weights of words; W_{len} indicates the word length weights of words; and W_{wem} indicates the semantic weights of words.

3. Results

In order to verify the effectiveness of the IWF-TextRank algorithm, a comparative analysis was carried out of the extraction effects of different feature conditions, different parameter conditions, and different algorithms, respectively.

3.1. Evaluation Indicators

Keyword extraction generally adopts accuracy (P), recall (R), and F-value as the evaluation index of extraction effects [31]. Accuracy and recall affect each other: the higher the accuracy, the lower the recall, so there is a contradiction between accuracy and recall, which can be weighted and reconciled by the F-value. The F-value is the result of the comprehensive consideration of the *p*-value and the R-value, and the higher the F-value, the better the effectiveness of the experimental method. The formula for this is shown below.

$$P = \frac{\sum_{i=1}^{N} \frac{|A_i \cap B_i|}{|A_i|}}{N} \tag{14}$$

$$R = \frac{\sum_{i=1}^{N} \frac{|A_i \cap B_i|}{|B_i|}}{N} \tag{15}$$

$$F = \frac{2PR}{P+R} \tag{16}$$

where A_i denotes the set of keywords extracted in the *ith* document; B_i denotes the set of manually labelled keywords in the ith document; and N denotes the number of documents in the test document set.

3.2. Parameter Setting

In this experiment, the accident causation data from the 2020 traffic accident data of a certain city was used as the dataset, and a total of 400 documents were obtained after processing. The data are provided by relevant departments and have high authenticity and accuracy. The dataset covers 4419 traffic accident records, contains rich accident information, and has a high value density. The data records are divided into two categories: 4364 data are recorded in table form, and another 55 accident data are recorded in text form. In general, the information in the dataset includes the specific time, location, weather conditions, cause of accident, type of accident, relevant information regarding the accident participants, vehicle information, and other fields. The experimental process is divided into the following two parts:

- Parameter training: 50% of the data is selected as the training set, which is used to obtain the optimal combination of feature parameters.
- Keyword extraction effect comparison: The remaining 50% is selected to test the keyword extraction effect of different algorithms and compared. In order to facilitate

the comparison, the documents after word splitting are manually annotated, and each document is annotated with 10 words as the annotated keywords.

The initial keyword set is obtained by preprocessing the data, after which the statistical algorithms are used to count the word frequency, lexicality, and word length of the words, respectively. The statistical results are shown in Tables 4 and 5.

Table 4. Results of lexical statistics.

Part of Speech (Noun, Verb, Adjective Etc.)	Prepositions	Noun (Part of Speech)	Adverb	Conjunctions
Number of words	5988	5596	361	0

Table 5. Results of word length statistics.

Word Length Scale	1	2	3	4	>4
Number of words	96	8056	265	3528	0

Based on the results of lexical statistics, verbs and nouns are selected as secondary features. According to the number of words contained in the lexical properties, it can be determined that the importance of verbs is greater than that of nouns, which will be recorded as B_1 and B_2, respectively, and then $B_1 > B_2$. From the results of the word length statistics, it can be seen that words with a word length of 2 and 4 account for a larger proportion of the word length, giving word length rankings of 2, 4, 3, >4, and 1, which will be recorded as C_1, C_2, C_3, C_4, and C_5, respectively.

According to the above statistics, the importance of the word and the word length ranking can be determined, and the order relationship analysis method can be used to obtain the weight of each indicator. The final weights of the indicators are shown in Table 6.

Table 6. Table of feature weights.

Weighting at the First Level		Secondary Weights	
Serial number	Weight	Serial number	Relative weight
A: Word frequency	0.342	A_1	TFIDF value
B: Lexical	0.285	B_1 B_2	0.524 0.475
C: Word length	0.204	C_1 C_2 C_3 C_4 C_5	0.384 0.213 0.152 0.141 0.122
D: Semantics	0.169	D_1	Wsim

After calculating the word feature weights, the BP neural network is used to calculate the weight assignment parameters of the word, the input of the neural network, and whether it is a keyword as the output (where 1 means it is a keyword, and 0 means it is not a keyword). The main parameters for the training of the BP neural network model are as follows: n-samples = 2000, noise = 0.4, random state = None, max epochs = 1000, learn rate = 0.035. The final obtained values for α, β, γ, and δ are 0.33, 0.35, 0.21, and 0.11.

3.3. Comparison of Results

3.3.1. Comparison of Results for Different Features

(1) Comparison of results for single features

The keyword extraction experiment under single-feature conditions can not only be used to compare and analyse the results of multi-factor and single-factor extraction, but can also preliminarily verify which feature enhancement can more effectively improve the accuracy of keyword extraction. The feature parameter settings are shown in Table 7, in which groups 1–4 represent the parameter settings when the single feature is word frequency, word nature, word length, and semantics, respectively, and group 9 is the parameter settings under the selected feature conditions in this paper. The extraction results are shown in Table 8 and Figure 3, where (a), (b), and (c) represent the results for the *p*-value, R-value, and F-value under different conditions, respectively.

Table 7. Single-feature parameter settings.

Serial Number	Diagnostic Property	α	β	γ	δ
1	Word frequency	1.00	0.00	0.00	0.00
2	Part of speech (noun, verb, adjective etc.)	0.00	1.00	0.00	0.00
3	Word length	0.00	0.00	1.00	0.00
4	Meaning of words	0.00	0.00	0.00	1.00
9	Word frequency—lexicality—word length—semantics	0.33	0.35	0.21	0.11

Table 8. Table of single-feature results.

Arithmetic	Accuracy (P)				Recall (R)				F-Value			
	3	5	7	10	3	5	7	10	3	5	7	10
1	0.713	0.683	0.631	0.592	0.207	0.338	0.438	0.562	0.321	0.452	0.518	0.576
2	0.719	0.687	0.637	0.599	0.211	0.340	0.440	0.566	0.326	0.454	0.520	0.582
3	0.702	0.667	0.623	0.585	0.205	0.334	0.434	0.557	0.317	0.445	0.511	0.580
4	0.694	0.649	0.612	0.564	0.202	0.325	0.428	0.551	0.313	0.433	0.504	0.557
9	0.732	0.704	0.652	0.613	0.229	0.357	0.456	0.584	0.348	0.473	0.537	0.598

As can be seen from the figure, when the number of extracted keywords is 3, 5, 7, or 10, respectively, the extraction effect under the condition of multi-feature combination proposed in this paper is significantly better than that under the condition of a single feature; the comparative analysis of the extraction effect between individual features shows that, among them, lexicality is the optimal result for the feature, followed by word frequency, and then word length. Its *p*-value, R-value, and F-value are significantly higher than those when semantics are used as a single feature, indicating that word frequency and lexical features have a greater impact on the keyword extraction effect. The word length feature has a relatively smaller impact on the keyword extraction effect, while the extraction effect is the worst when the algorithm is improved only with semantics as a feature; thus, semantic features have a smaller impact on the extraction result, which also confirms the results of parameter training from the side.

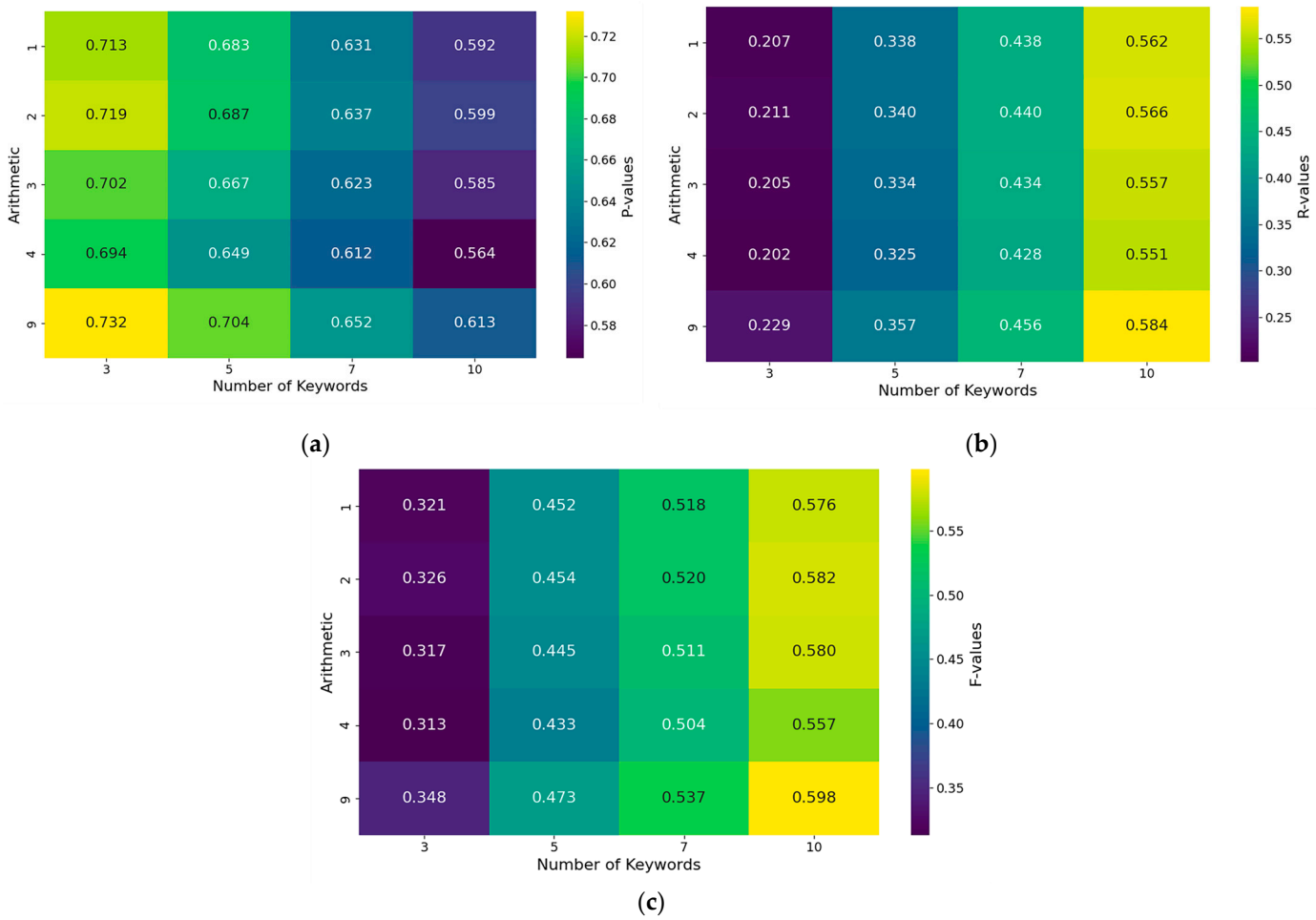


Figure 3. Comparison of results for single features. (**a**) Comparison of *p*-values for single features; (**b**) comparison of R-values for single features; (**c**) comparison plot of F-values for single features.

(2) Comparison of results for multiple features

According to the results of the single-feature experiments, it can be seen that lexicality has a greater impact on the keyword extraction results, followed by word frequency, and then word length, while semantic features have a smaller impact. Therefore, the lexicality, word frequency, and word length features that have a greater impact were selected to carry out the double-feature combination experiments and triple-feature combination experiments, respectively, and the results were compared and analysed with the results of the textual feature conditions. In the double-feature experiment, the feature weights were obtained using the sequential relationship analysis method; the degree of importance between the indicators was lexicality > frequency > length, and the ratio of the degree of importance was lexicality/frequency/length 1:1.3, lexicality/length 1:1.5, and frequency/length 1:1.2, which were then calculated to obtain the weights of the features and inputted into the BP neural network training to obtain the feature parameters. In the three-feature experiment, word/word frequency/word length was 1.3:1.2, and the weights were calculated in the same way as in the two-feature experiment. The specific parameter settings are shown in Table 9, and the experimental results are shown in Table 10 and Figure 4, where the red curves represent the extraction results under the conditions of the feature parameters in this paper.

Table 9. Multi-feature parameter settings.

Serial Number	Diagnostic Property	α	β	γ	δ
5	Word Frequency—Word Properties	0.41	0.59	0.00	0.00
6	Word Frequency—Word Length	0.56	0.00	0.44	0.00
7	Lexical Category—Word Length	0.00	0.62	0.38	0.00
8	Word Frequency—Word Length	0.32	0.41	0.26	0.00
9	Word Frequency—Lexicality—Word Length—Semantics	0.33	0.35	0.21	0.11

Table 10. Multi-feature results.

Arithmetic	Accuracy (P)				Recall (R)				F-Value			
	3	5	7	10	3	5	7	10	3	5	7	10
5	0.725	0.696	0.647	0.608	0.220	0.350	0.448	0.577	0.337	0.466	0.529	0.592
6	0.720	0.689	0.639	0.601	0.214	0.342	0.441	0.569	0.329	0.457	0.521	0.584
7	0.723	0.692	0.642	0.605	0.216	0.346	0.443	0.573	0.333	0.461	0.525	0.589
8	0.728	0.699	0.649	0.610	0.225	0.353	0.451	0.580	0.344	0.469	0.534	0.595
9	0.732	0.704	0.652	0.613	0.229	0.357	0.456	0.584	0.348	0.473	0.537	0.598

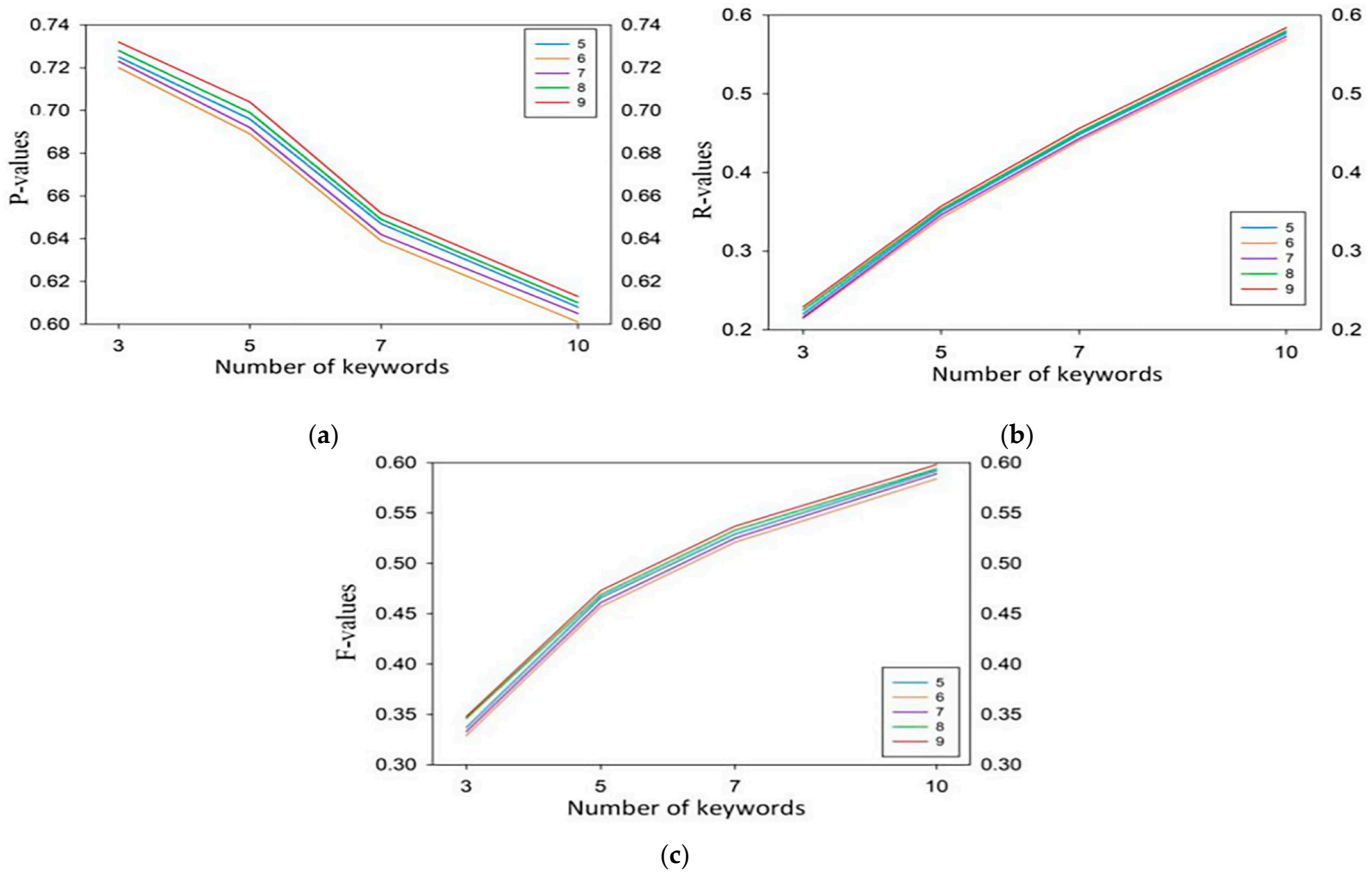


Figure 4. Comparison of multiple features. (**a**) Comparison of *p*-values for multiple features; (**b**) comparison of R-values for multiple features; (**c**) comparison F-values for multiple features.

As can be seen from the figure, the results under the combination of features in this paper are significantly better than the results under the combination of double and triple features, and the extraction effect is better than the combination of two and two when combining word frequency, word length, and word nature; a comparative analysis of the

results between the combination of double features reveals that the extraction effect is the best when combining word frequency and word nature.

The above experimental comparison found that the *p*-value, R-value, and F-value, under the conditions of the features selected in this paper, were greater than the other feature combinations, indicating that the features selected in this paper are more suitable for the text dataset.

3.3.2. Comparison of Results for Different Parameters

In order to illustrate the effect of keyword extraction under the parameter conditions of this paper, five groups of experiments were set up. Among them, group 1 considers the four features as equally important; group 5 is the combination of parameters obtained from the training of this paper; and the remaining combinations are fine-tuned on the basis of the parameters obtained from the training. The specific parameter settings are shown in Table 11. The results for the *p*-value, R-value, and F-value under different parameter conditions are shown in Table 12 and Figure 5, and the red curves in the figure represent the extraction results under the conditions of feature parameters in this paper.

Table 11. Parameter settings.

Serial Number	α	β	γ	δ
1	0.25	0.25	0.25	0.25
2	0.31	0.34	0.22	0.13
3	0.34	0.35	0.20	0.11
4	0.35	0.35	0.19	0.11
5	0.33	0.35	0.21	0.11

Table 12. Table of results for different parameters.

Arithmetic	Accuracy (P)				Recall (R)				F-Value			
	3	5	7	10	3	5	7	10	3	5	7	10
1	0.707	0.671	0.641	0.601	0.212	0.336	0.430	0.553	0.326	0.448	0.516	0.569
2	0.728	0.699	0.644	0.606	0.217	0.348	0.442	0.571	0.334	0.464	0.528	0.585
3	0.729	0.701	0.647	0.609	0.221	0.351	0.448	0.576	0.339	0.467	0.530	0.592
4	0.730	0.703	0.650	0.611	0.225	0.354	0.453	0.581	0.343	0.471	0.534	0.595
5	0.732	0.704	0.652	0.613	0.229	0.357	0.456	0.584	0.348	0.473	0.537	0.598

As can be seen from the figure, after fine-tuning each parameter on the basis of the parameters obtained from training, the extraction effect is optimal under the parameter conditions where this paper is located, which indicates that α, β, γ, δ is the optimal parameter combination when α, β, γ, and δ are 0.33, 0.35, 0.21, and 0.11, and at this time, the improved algorithm has the best extraction effect.

3.3.3. Comparison of Results of Different Extraction Methods

In order to further verify the extraction effect of the IWF-TextRank algorithm, it was compared with that of the traditional TextRank algorithm as well as the TFIDF algorithm, and the extraction results are shown in Table 13 and Figure 6.

Table 13. Table of results for different algorithms.

Arithmetic	Accuracy (P)				Recall (R)				F-Value			
	3	5	7	10	3	5	7	10	3	5	7	10
TFIDF	0.685	0.632	0.601	0.543	0.189	0.284	0.399	0.536	0.296	0.392	0.480	0.539
TextRank	0.692	0.643	0.608	0.557	0.201	0.297	0.406	0.545	0.311	0.406	0.487	0.551
IWF-TextRank	0.732	0.704	0.652	0.613	0.229	0.357	0.456	0.584	0.348	0.473	0.537	0.598

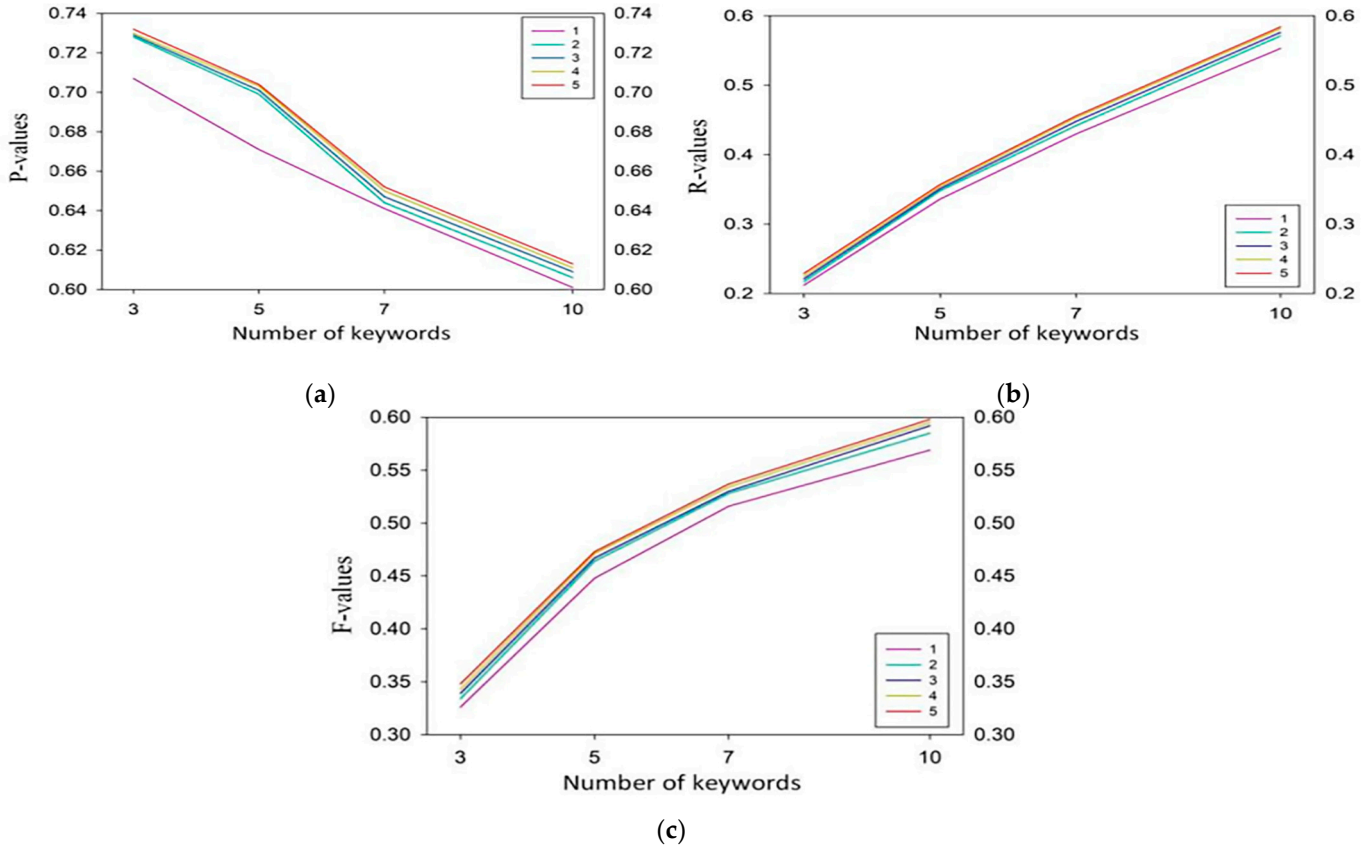


Figure 5. Comparison of different parameters. (**a**) Comparison of *p*-values for different parameters; (**b**) comparison of R-values for different parameters; (**c**) comparison of F-values for different parameters.

As can be seen from the figure, when the number of extracted keywords is 3, 5, 7, 10, the extraction effect of the TFIDF algorithm based on word frequency consideration is relatively poor compared to the other two methods; relative to the TFIDF algorithm, the keyword extraction effect of the traditional TextRank algorithm has been improved to a certain extent, and the extraction effect of the IWF-TextRank algorithm is significantly better than the other two methods. Specifically, there is a 10.06% improvement in accuracy over TextRank and an 7.16% improvement in recall when the number of keywords is 10. These enhancements were validated by a statistical significance test ($p < 0.05$), indicating that the observed improvements are statistically robust and not due to random variation.

The integration of semantic features with the BP neural network plays a pivotal role in enhancing the algorithm's performance. By capturing deeper contextual relationships between terms, the BP neural network optimises the weight distribution of features, leading to the more accurate identification of domain-specific keywords. Furthermore, the improvement in the F-value demonstrates that the IWF-TextRank algorithm effectively balances the inherent trade-off between precision and recall, resulting in a more comprehensive and reliable keyword extraction process. This balance is particularly advantageous for traffic accident data analysis, where the precise extraction of relevant keywords is critical for accurate summarisation and further analytical tasks.

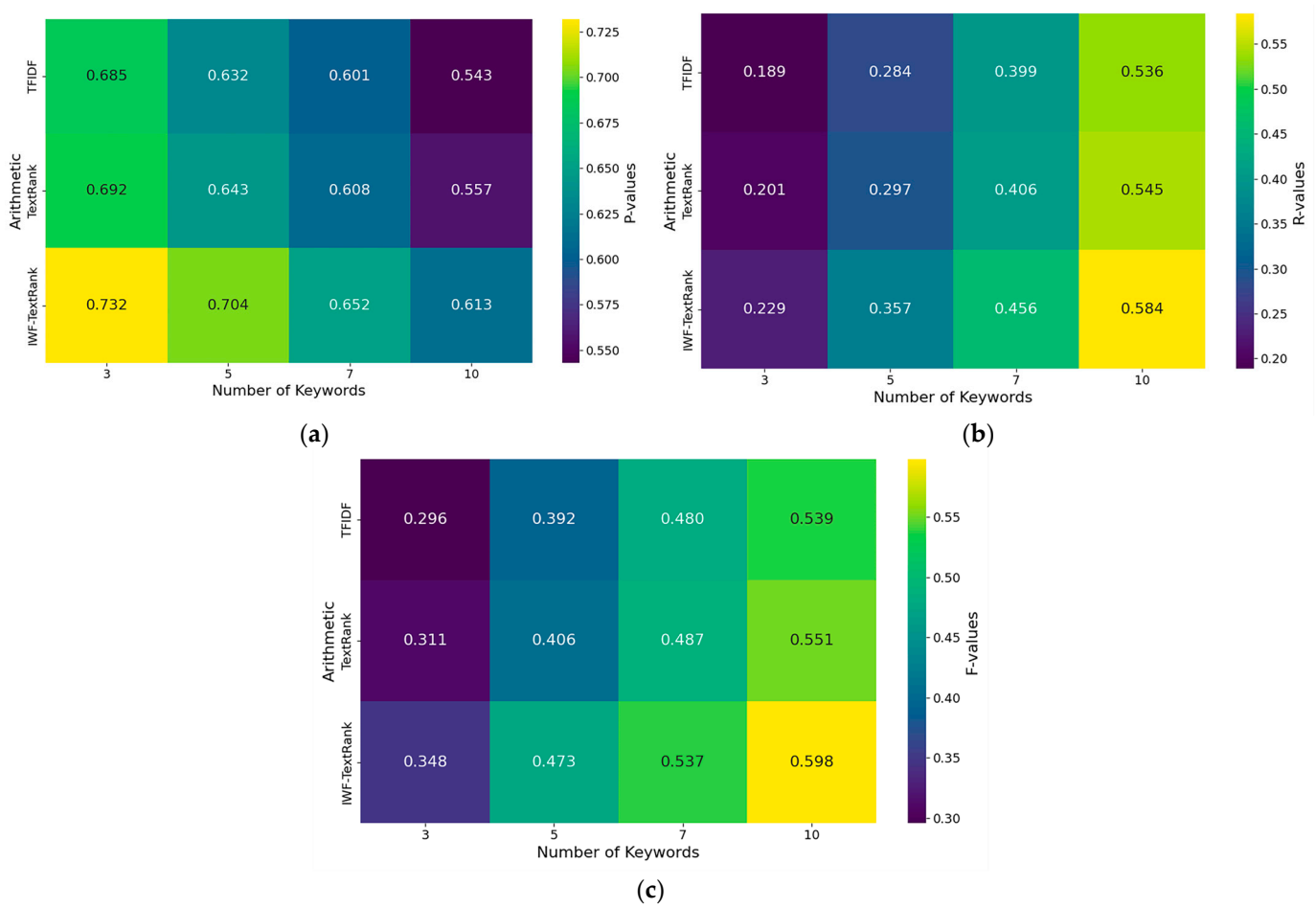


Figure 6. Comparison of different extraction methods. (**a**) Comparison of *p*-values of different extraction methods; (**b**) comparison of R values of different extraction methods; (**c**) comparison of F-values of different extraction methods.

4. Discussion

The proposed IWF-TextRank algorithm demonstrates significant improvements in keyword extraction for traffic accident texts, primarily due to three key enhancements: multi-feature weighting, BP neural network optimisation, and semantic enhancement through the CBOW model. Each of these factors is summarised below with supporting quantitative evidence.

1. Multi-Feature Weighting Mechanism

Unlike the traditional TextRank, which relies solely on word frequency, the IWF-TextRank incorporates additional features such as lexicality and word length to provide a more comprehensive weighting for each candidate keyword. The experimental results indicate that the inclusion of these features improves both the precision and recall of keyword extraction by approximately 7%, particularly by accurately capturing the relevance of nouns and verbs in accident descriptions. Additionally, the word-length feature enables the model to better extract keywords with longer lengths, further improving the extraction of key information.

2. Optimisation through BP Neural Network

The BP neural network automatically balances the weights for word frequency, part of speech, and word length, eliminating subjective manual adjustments. The cross-validation results show that the BP optimisation enhances extraction accuracy with statistical signif-

icance ($p < 0.05$), indicating the network's effective contribution to model stability and precision in identifying critical accident-related keywords.

3. Semantic Enhancement via the CBOW Model

Integrating the CBOW model from Word2Vec allows the algorithm to capture deeper semantic relationships between words.

Despite the advantages of IWF-TextRank in keyword extraction, certain limitations remain. First, the dataset is limited to traffic accident records from a single city in 2020, potentially limiting the model's generalisability across regions and timeframes. Second, while BP neural network optimisation performs well on small datasets, it may require additional tuning for larger datasets. Finally, although the CBOW model improves semantic understanding, it may face challenges with longer or multi-topic texts.

5. Conclusions

This study presents a new keyword extraction algorithm, IWF-TextRank, which combines multi-feature weighting, BP neural network optimisation, and the CBOW model from Word2Vec to achieve enhanced performance in domain-specific datasets like traffic accident reports. The main contributions of this work are as follows:

- A multi-feature weighting mechanism incorporating word frequency, lexicality, and word length, enabling more accurate identification of core keywords;
- BP neural network optimisation for automated weight distribution, reducing subjectivity and improving stability and precision in keyword extraction;
- Integration of the CBOW model, enhancing the model's semantic understanding and enabling deeper contextual associations for more accurate keyword identification.

These contributions result in significant improvements in keyword extraction accuracy and robustness within traffic accident analysis, providing a useful framework for similar applications.

Future work could involve expanding the dataset to other regions and years to validate the robustness of IWF-TextRank, combining it with more advanced deep learning models such as Transformers for complex text analysis, and exploring its potential in other specialised fields (e.g., medical or legal documents). Further research could also investigate automated parameter adjustments to improve adaptability across diverse datasets.

Author Contributions: Conceptualisation, L.Z. and W.W.; methodology, W.W.; software, L.Z.; validation, L.Z., W.W. and Y.W.; formal analysis, J.M.; investigation, Y.W.; resources, J.M.; data curation, W.W.; writing—original draft preparation, W.W.; writing—review and editing, L.Z.; visualisation, W.W.; supervision, J.M.; project administration, L.Z.; funding acquisition, J.M. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the support of the participating organisation and its personnel who provided various effective assistance. The authors also sincerely thank the financial support provided by the abovementioned mechanism. This research was funded by the Postgraduate Research and Practice Innovation Program of Jiangsu Province (SJCX20_1117, SJCX21_1420 and KYCX21_2999), the Construction System Project of Jiangsu Province (2020ZD14 and 2018ZD258), Suzhou Social Science Fund (Y2020LX017 and Y2020LX025), and the Philosophy and Social Science Projects of Universities in Jiangsu Province (2018SJA1348 and 2023SJYB1420).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available on request to the corresponding author.

Acknowledgments: Thanks are extended to the relevant departments for providing the 2020 traffic accident data so that the research data in this paper is true and accurate.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yu, B.G.; Zhang, H.M.; Cao, Y.M. *TextRank Keyword Extraction Method Based on Multivariate Feature Weighting*; Digital Library Forum: Beijing, China, 2020; pp. 41–50. [CrossRef]
2. Qiu, Q.; Xie, Z.; Wu, L.; Li, W. Geoscience keyphrase extraction algorithm using enhanced word embedding. *Expert Syst. Appl.* **2019**, *125*, 157–169. [CrossRef]
3. Bennani-Smires, K.; Musat, C.; Hossmann, A.; Baeriswyl, M.; Jaggi, M. Simple unsupervised keyphrase extraction using sentence embeddings. *arXiv* **2018**, arXiv:1801.04470.
4. Habibi, M.; Popescu-Belis, A. Keyword extraction and clustering for document recommendation in conversations. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 746–759. [CrossRef]
5. Xie, F.; Wu, X.D.; Zhu, X.Q. Document-specific keyphrase extraction using sequential patterns with wildcards. In Proceedings of the 2014 IEEE International Conference on Data Mining, Jinan, China, 4–17 December 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1055–1060. [CrossRef]
6. Chang, Y.C.; Zhang, Y.X.; Wang, H.; Wan, H.Y.; Xiao, C.J. A Review of Feature-Driven Keyword Extraction Algorithms. *J. Softw.* **2018**, *29*, 2046–2070. [CrossRef]
7. Mihalcea, R.; Tarau, P. Textrank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; pp. 404–411.
8. Ding, S.; Su, C.; Yu, J. An optimizing BP neural network algorithm based on genetic algorithm. *Artif. Intell. Rev.* **2011**, *36*, 153–162. [CrossRef]
9. Bai, F.B.; Chang, L.; Wang, S.F.; Li, B.; Wang, Y.J.; Zhou, H.; Liu, Y. Research on improved method of keyword extraction for adjudication documents. *J. Comput. Eng. Appl.* **2020**, *56*, 153. [CrossRef]
10. Zhang, J. Intelligence keyword extraction method based on improved TF-IDF algorithm. *J. Intell.* **2014**, *33*, 153–155.
11. Wang, Z.; Dong, W.A.; Qing, L.I. Keyword Extraction from Scientific Research Projects Based on SRP-TF-IDF. *Chin. J. Electron.* **2021**, *30*, 652–657. [CrossRef]
12. Mihunov, V.V.; Jafari, N.H.; Wang, K.; Lam, N.S.; Govender, D. Disaster impacts surveillance from social media with topic modeling and feature extraction: Case of hurricane harvey. *Int. J. Disaster Risk Sci.* **2022**, *13*, 729–742. [CrossRef]
13. Qiu, Q.; Tian, M.; Tao, L.; Xie, Z.; Ma, K. Semantic information extraction and search of mineral exploration data using text mining and deep learning methods. *Ore Geol. Rev.* **2024**, *165*, 105863. [CrossRef]
14. Guauque-Olarte, S.; Cifuentes, C.L.; Fong, C. Oral manifestations in patients with coronavirus disease 2019 (COVID-19) identified using text mining: An observational study. *Sci. Rep.* **2023**, *13*, 17770. [CrossRef] [PubMed]
15. Im, Y.; Park, J.; Kim, M.; Park, K. Comparative study on perceived trust of topic modeling based on affective level of educational text. *Appl. Sci.* **2019**, *9*, 4565. [CrossRef]
16. Li, M.C.; Lu, Y.G.; Tian, D.; Shen, Y. Intelligent analysis of hydropower project progress management text based on improved LDA. *J. Hydropower Gener.* **2022**, *41*, 133–141. [CrossRef]
17. Wang, X.X.; Han, B.; Gao, R.; Chen, P. Automatic Extraction of Text Summaries Based on Improved TextRank. *Comput. Appl. Softw.* **2021**, *38*, 155–160. [CrossRef]
18. Xu, X.T.; Chai, X.L.; Xie, B.; Shen, C.; Wang, J.P. Chinese text summary extraction based on improved TextRank algorithm. *Comput. Eng.* **2019**, *45*, 273–277. [CrossRef]
19. Wan, X.; Xiao, J. Graph-Based Keyphrase Extraction Using Cross-Sentence Context. *Nat. Lang. Eng.* **2021**, *27*, 1–20.
20. Lopez, L.; Toledo, J.; Carrasco, R.; Pinzón, H. A Neural Network Model for Keyphrase Extraction Based on Word Embeddings and Linguistic Features. *Appl. Sci.* **2020**, *10*, 2430.
21. Boudin, F. Unsupervised Keyphrase Extraction with Multipartite Graphs. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, New Orleans, LA, USA, 1–6 June 2018; pp. 667–672.
22. Wu, X.; Kumar, V.; Ross Quinlan, J.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G.J.; Ng, A.; Liu, B.; Yu, P.S.; et al. Top 10 algorithms in data mining. *Knowl. Inf. Syst.* **2008**, *14*, 1–37. [CrossRef]
23. Ai, J.Y. A Study on TextRank Keyword Extraction Method for Tibetan Text with Multi-Feature Fusion. *Intell. Explor.* **2020**, *7*, 1–6. [CrossRef]
24. Zhang, L.J.; Li, Y.L.; Zeng, Q.T.; Lei, J.L.; Yang, P. Keyword extraction algorithm based on improved TextRank. *J. Beijing Inst. Print.* **2016**, *24*, 51–55. [CrossRef]
25. Ghorpade, S.; Khan, A.; Chaurasia, A.; Rao, V.; Chhabria, A. A Comparative Analysis of TextRank and LexRank Algorithms Using Text Summarization. In *Proceedings of the International Joint Conference on Advances in Computational Intelligence—IJCACI 2022*; Uddin, M.S., Bansal, J.C., Eds.; Algorithms for Intelligent Systems; Springer: Singapore, 2024. [CrossRef]
26. Liu, Q.Q. Application of Improved TFIDF Algorithm in Text Analysis. Master's Thesis, Nanchang University, Nanchang, China, 2019.
27. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
28. Huang, R. Research on Content-Based and Word2vec-Based Catechism Recommendation Algorithm. Master's Thesis, Shandong Normal University, Jinan, China, 2019.
29. Granberg, C. *Character Animation with Direct3D*; Charles River Media: New York, NY, USA, 2013; 448p.

30. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef] [PubMed]
31. Barman, U.; Barman, V.; Choudhury, N.K.; Rahman, M.; Sarma, S.K. Unsupervised Extractive News Articles Summarization leveraging Statistical, Topic-Modelling and Graph-based Approaches. *J. Sci. Ind. Res.* **2022**, *81*, 952–962. [CrossRef]