Information Dynamics and Applications
https://www.acadlore.com/journals/IDA

An Improved TextRank Keyword Extraction Method Based on the Watts-Strogatz Model

Aofan Li[1], Lin Zhang[2*], Ashim Khadka[3]

[1] Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, 223003 Huaian, China
[2] Faculty of Management Engineering, Huaiyin Institute of Technology, 223003 Huaian, China
[3] Nepal College of Information Technology, Pokhara University, 44700 Lalitpur, Nepal

[*] Correspondence: Lin Zhang(zlmjl@hyit.edu.cn)

Received: 01-10-2024 **Revised:** 04-17-2024 **Accepted:** 04-26-2024

Citation: A. F. Li, L. Zhang, and A. Khadka, "An improved textrank keyword extraction method based on the Watts-Strogatz model," *Inf. Dyn. Appl.*, vol. 3, no. 2, pp. 77–88, 2024. https://doi.org/10.56578/ida030201.

Abstract: Traditional methods for keyword extraction predominantly rely on statistical relationships between words, neglecting the cohesive structure of the extracted keyword set. This study introduces an enhanced method for keyword extraction, utilizing the Watts-Strogatz model to construct a word network graph from candidate words within the text. By leveraging the characteristics of small-world networks (SWNs), i.e., short average path lengths and high clustering coefficients, the method ascertains the relevance between words and their impact on sentence cohesion. A comprehensive weight for each word is calculated through a linear weighting of features including part of speech, position, and Term Frequency-Inverse Document Frequency (TF-IDF), subsequently improving the impact factors of the TextRank algorithm for obtaining the final weight of candidate words. This approach facilitates the extraction of keywords based on the final weight outcomes. Through uncovering the deep hidden structures of feature words, the method effectively reveals the connectivity within the word network graph. Experiments demonstrate superiority over existing methods in terms of precision, recall, and F1-measure.

Keywords: Watts-Strogatz model; Word network graph; Cohesive structure; TextRank++

1 Introduction

Keywords, defined as single or multiple word units, are employed to express the central theme of a text [1]. The extraction of keywords plays a pivotal role in applications such as information retrieval, text summarization, automatic categorization, clustering, and indexing [2]. Manual extraction of keywords is both labor-intensive and time-consuming. Hence, the automation of keyword extraction has garnered the attention of researchers. Since the inception of a keyword extraction method based on word frequency statistical rules by Luhn [3] in 1957, the development of automatic keyword extraction technologies has evolved over more than seven decades, spawning a multitude of methods. These methods are principally categorized into supervised and unsupervised extraction methods [4]. Although supervised keyword extraction methods can achieve higher extraction accuracy, they require a large volume of annotated data, incurring significant manual labor costs. In contrast, unsupervised methods are less demanding regarding data requirements. They eliminate the need for manually creating and maintaining a lexicon, as well as for manual standard corpus assistance in training, thereby making unsupervised keyword extraction techniques more widely applicable. Notably, unsupervised keyword extraction includes methods based on statistical features, graph models, and word networks, among which the methods based on word graphs exhibit superior performance in capturing textual structure and relationships.

Typically, graphs are constructed based on words and their relationships, followed by the determination of word importance through metrics such as centrality and frequency. However, the concept of "relationships" in most of these methods is solely based on the relevance of words, i.e., co-occurrence within a fixed window size or semantic relevance derived from external knowledge bases [5, 6]. Consequently, these methods fail to consider the impact of the deeper structural attributes of words, potentially leading to the loss of significant information. Moreover, the keywords extracted using graph-based methods in most studies are merely individually relevant, without reflecting the cohesive structure among keywords. Therefore, identifying a method to assess the internal relevance among keywords remains a major challenge in keyword extraction research.

SWNs are mathematical graphs in which the majority of nodes are not adjacent to each other, yet neighbors of any given node are likely to be neighbors themselves. Moreover, most nodes can be reached from any other through a small number of steps or hops. The Watts-Strogatz modelproposed by Watts and Strogatz [7] positions itself in their 1998 paper published in Nature, positions itself between regular and random networks. It is characterized by short average path lengths and high clustering coefficients, partially explaining the "small-world" phenomenon observed in many networks. The presence of small-world properties in human language has been demonstrated by Cancho and Solé [8], suggesting that words in a word network with high clustering and short average distances represent central words in sentences. Thus, identifying words with these characteristics provides a profound basis for accurately extracting keywords. For instance, research conducted by Ma et al. [9] on Chinese document keyword extraction utilizing the SWN model, which employs changes in word clustering coefficients and average shortest path lengths to measure word importance, further confirmed the effectiveness of using SWNs for keyword extraction. However, this method did not consider the intrinsic features of words in the process of keyword extraction.

In light of these issues, this study introduces a new method for keyword extraction that employs the Watts-Strogatz model to construct a word network graph. The average path length of the graph reflects the relatedness between words, while the nodes' clustering coefficient indicates the cohesive structure among words. These two SWN properties are used to signify the importance of words. By integrating part-of-speech, position, and TF-IDF features, a comprehensive weight for each word is determined and applied to refine the impact factors in the TextRank algorithm's candidate word graph, optimizing the iterative calculation process of words in the graph. This study considers both the relatedness and cohesive structure among words, proposing an improved TextRank algorithm. The effectiveness of this method in extracting keywords from text is validated through experimentation.

2 Related Work

2.1 Keyword Extraction

Unsupervised keyword extraction methods are primarily divided into those based on statistics, themes, deep learning, and graphs.

(a) Methods based on statistical features. Keyword extraction leveraging statistical features involves the analysis of specific indicators for candidate words, followed by ranking these candidates based on statistical outcomes. This approach requires neither training data nor external knowledge bases. In 1958, Luhn [3] introduced a method for extracting keywords based on Term Frequency (TF), suggesting that higher-TF words are more likely to be keywords of the document. In 1972, Sparck Jones [10] proposed the concept of Inverse Document Frequency (IDF), positing that if fewer documents contain a certain feature word, resulting in a higher IDF, this indicates the feature word's strong discriminative power between categories. Building on this, Salton and Buckley [11] combined TF with IDF to introduce the TF-IDF method. This method selects words with the highest TF-IDF values as the document's keywords and remains widely used to this day. Subsequently, Xuan and Wei [12] integrated word part-of-speech and the position of the first appearance with TF-IDF, proposing the Term Frequency Rate-IDF (TFR-IDF), which achieved better results compared to the traditional TF-IDF. Ding and Wang [13] added feature word category, part of speech, and text position information to the original structure of TF-IDF, developing a multi-factor weighted method, TF-IDF-PPC. Bounabi et al. [14] introduced a new term weighting method based on TF-IDF, i.e., Neutral Term Frequency-IDF (NTF-IDF). This method defines term relevance and ambiguity using term frequency as a component and considers the application of natural language (NL) in weight inference as a baseline model. It corrects the shortcomings of the fuzzy logic-based FTF-IDF and enhances the accuracy and relevance of the resulting feature vectors. Although methods based on statistics are simple and quick, capable of extracting document keywords through high-frequency words, they tend to overlook words that are significant to the document but not frequently occurring, hence providing an incomplete extraction result.

(b) Methods based on graph models. In the keyword extraction based on graph models, candidate words within documents are viewed as nodes, with the textual content forming a word graph. Connections between nodes are established following specific rules, and keywords are identified by calculating and ranking the weight of each node. TextRank [15] stands as a quintessential example within the methods based on graph models, calculating connections between words based on their co-occurrence within the document. This method, requiring no prior training across multiple documents and celebrated for its simplicity and effectiveness, has seen widespread use. However, a limitation of this approach is its assumption of a relationship between any two candidate words appearing within the same window, without differentiating the strength of such connections. In 2008, Wan and Xiao [16] introduced Single Rank, measuring the weight of edges between nodes by the co-occurrence count of words. In 2015, SGRank [17] assigned values to the edges of candidate words using statistical indicators such as the position of the first occurrence and word length. The PositionRank algorithm, proposed by Florescu and Caragea [18] in 2017, incorporated the position information of words into the calculation of edge weights. In 2019, Dong et al. [19] integrated co-occurrence features, span features, and text theme features of words with the TextRank method, building on the basis of word position characteristics. Addressing the shortcomings of the TextRank method, which only considers the co-occurrence and initial significance of words when extracting keywords, Qiu and Zheng [20] proposed an unsupervised keyword extraction method based on the tolerance rough set. This method characterizes the initial importance of words through the calculation of word membership within each document, forming a fuzzy membership degree matrix. The methods based on graph models, to a certain extent reliant on co-occurrence or syntactic relationships, still lack a deeper understanding of the relevance and structure between words, failing to capture the document's contextual complexity. Thus, there is a need to explore graphs with more complex forms of relevance.

(c) Methods based on word networks. Keyword extraction based on word networks involves establishing a document's word network model, where words within the document serve as nodes and some form of relationship between words forms the edges, with node importance based on average distance length or clustering coefficient. Word networks can reflect not only the content of the document but also its overall and semantic structure. The keyword algorithm [21], first introduced to represent the main content of documents through word networks, is built upon TF statistics. This algorithm maps words and their semantic relations onto a word network graph, with network vertices representing words in the document and edges representing the relationships between words, allowing the construction of the document's word network based on the connectivity among words. Zhang et al. [22], building on TF statistics, constructed an undirected word network with n vertices, utilizing the betweenness centrality (BC) metric to quantify the importance of nodes and extract several significant vertices as the document's keywords. To address the issue of sparsity and poor performance in short text data, Yang et al. [23] proposed a Word-Concept Heterogeneous Graph Convolutional Network model (WC-HGCN), which classifies short texts by introducing interaction information between words and concepts. Furthermore, following the demonstration of small-world properties in human language by Cancho and Solé [8], Zhu et al. [24] discovered small-world properties in document structures and proposed the SWS(Small World Structure) for constructing the overall structure graph of documents. On this basis, the SWN_L algorithm(The Simple-L mining algorithm based on SWN) was proposed to apply small-world theory to the extraction of document keywords. Dong [25] analyzed and summarized the SWS and SWN_L algorithms, proposing a new algorithm for constructing the overall structure graph of documents and a complex keyword mining algorithm for documents, offering new methods for the automatic analysis of document semantics.

2.2 The TextRank Algorithm Based on Graph Models

The principle of keyword extraction using the TextRank algorithm, based on word graph models, involves treating each word or sentence as a node within a grid graph. These nodes are iteratively calculated until convergence is achieved, resulting in specific weight values that are then sorted in descending order to identify the top N keywords.

The TextRank algorithm is an extension of the PageRank algorithm in both theory and application. PageRank determines the importance of a webpage by analyzing the links and relationships between pages. The TextRank algorithm, a variant of PageRank, can be understood through PageRank's concept as follows:

(a) A word that appears near many other words is considered important.

(b) A word with a high TextRank value increases the TextRank value of subsequent words.

The essence of the TextRank algorithm lies in treating sentences as nodes and adding weight to the edges between nodes. This weight represents the similarity value between two sentences, associating a word with the N words before and after it in the graph as adjacent words. Specifically, a sliding window of length N is set, within which all words are considered adjacent to the word node, essentially constructing a weighted undirected graph. The computation formula for the TextRank algorithm is as follows:

$$S(i) = (1 - d) + \sum_{j \in \mathrm{In}(j)} \frac{W_{ji}}{\sum_{k \in Out(j)} W_{jk}} S(j) \tag{1}$$

where, $G(V, E)$ represents the directed graphs with V as the set of vertices and E as the set of edges, In (i) denotes the set of all vertices pointing to vertex i, $Out(i)$ is the set of all vertices that points to i, d is the damping factor, and $S(i)$ represents the weight of the vertex i, which is set to 0.85 by convention.

In the TextRank algorithm, the importance of a target word increases with the number of words it co-occurs with, and accordingly, its weight increases. Moreover, the weight of each word node depends not only on its attributes but also on the attributes of all nodes it is connected to.

2.3 The Watts-Strogatz Model

Most real-world networks are neither entirely regular nor completely random, resembling social networks where individuals frequently interact with their neighbors but not solely with them, as in regular networks; individuals also have friends in distant locations. Transitioning from entirely regular to completely random networks, the Watts-Strogatz model [7] was proposed by Duncan J. Watts and Steven Strogatz in their 1998 paper published in Nature.

This model possesses the advantageous characteristics of high clustering and short average path lengths. These features make the SWNs more suitable for representing real networks than either completely regular or completely random networks.

The Watts-Strogatz model adopted in this study is positioned between regular and random networks. Its construction begins with a regular graph consisting of N nodes to create a nearest neighbor coupled network with N nodes. Each node is connected to $R/2$ neighbors on each side, where R is an even number. Edges in the network are randomly rewired with a probability p $(0 \leq p \leq 1)$; one end of an edge remains fixed while the other end is randomly reconnected to another node in the network, avoiding self-loops and parallel edges. As depicted in Figure 1, a value of p=0 corresponds to a completely regular network, and p=1 to a completely random network. The transition from a completely regular to a completely random network can be controlled by adjusting the probability p.

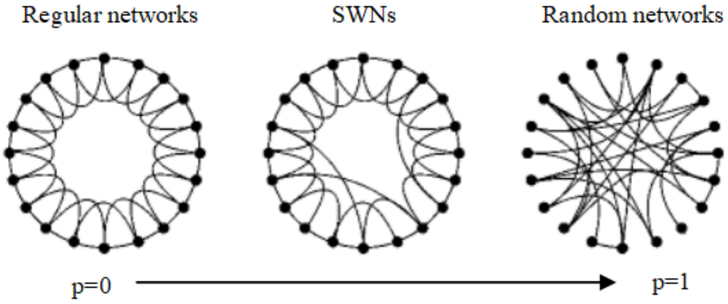


Figure 1. The Watts-Strogatz model

3 The WS-TextRank Algorithm

Building upon existing methods of keyword extraction, this study introduces an improved TextRank keyword extraction method based on the Watts-Strogatz model. This method utilizes the average path length of the WS SWNs to indicate the relatedness between words, and the clustering coefficient of nodes to reflect the contribution of words to sentences. Based on these two parameters, words are preliminarily marked for their importance. This is combined with word part-of-speech, position, and TF-IDF features alongside the TextRank method, considering both the attributes of words themselves and the relatedness and clustering among them.

3.1 Constructing the Word Network Graph

Let G be a graph, where the number of elements in the candidate word set corresponds to the number (N) of nodes in graph G. VG represents the set of nodes in graph G, where i $(i \in V_G)$ denotes any given node in graph G, and EG represents the set of edges in graph G, with E_{ij} denoting the edge connecting nodes i and j. Gi represents the subgraph of graph G, signifying the graph formed by all adjacent nodes directly connected to node i, with N_{Gi} being the number of nodes in subgraph Gi, and E_{Gi} representing the actual number of edges in subgraph Gi. Starting from a regular graph, a nearest neighbor coupled network comprising N nodes is constructed, where each node is connected to $R/2$ adjacent nodes on either side, with R being an even number. With a probability p $(0 \leq p \leq 1)$, every edge in the network is randomly reconnected, keeping one end of the edge fixed while selecting another node in the network randomly for the other end, with the stipulation that self-loops and parallel edges are not allowed.

(a) Clustering coefficient

In the word network graph, the clustering coefficient $C(i)$ of a node i is defined as the ratio of the actual number of edges in the subgraph Gi to the number of fully connected edges among the nodes in the subgraph. Clearly, $C(i)$ is a number between 0 and 1. The calculation method is as follows:

$$C(i) = \frac{2E_{Gi}}{N_{Gi}\left(N_{Gi}-1\right)} \tag{2}$$

The network's clustering coefficient $C(G)$ refers to the average value of the clustering coefficients of all nodes in the network, indicating the degree of clustering among the network's nodes. The clustering coefficient is a measure describing the extent to which nodes in a graph or network cluster together. The calculation method is as follows:

$$C(G) = \frac{1}{N}\sum_{i \in V_G} C(i) \tag{3}$$

In the word network graph, the closer a node i's clustering coefficient $C(i)$ is to 1, the more "clique-like" the tendency of the nodes surrounding it, contributing significantly to the network's clustering. This implies that the word represented by the node is of greater importance in the sentence. Generally, when the clustering coefficient $C(i)$ of a node i representing a word is large, the average $C(G)$ of the clustering coefficients of all nodes in the network is necessarily less than $C(i)$, meaning the ratio of $C(i)$ to the network's clustering coefficient $C(G)$ is greater than 1. Thus, the characteristics of important words must align with those nodes where this ratio exceeds 1. These nodes are marked as significant to simplify the complexity of extracting word features later. Therefore, the condition for assessing the importance of node clustering, C^*, is defined as follows:

$$C^* = \frac{C(i)}{C(G)}, C^* > 1 \tag{4}$$

(b) Average shortest path length

The journey from one node i to another node j via connected nodes, referred to as the path between the two points, where the shortest among these paths is also known as the distance between the two points, is denoted as d_{ij}. The shortest path length from a node to itself is defined to be 0. In the word network graph, the average path length $L(i)$ of a node i refers to the average value of the shortest path lengths between that node and all other nodes in graph G. The average shortest path length of node i is defined as follows:

$$L(i) = \frac{1}{N-1} \sum_{i,j \in V_G} d_{ij} \tag{5}$$

The average path length, also known as the characteristic path length or the average shortest path length, refers to the average of the shortest path lengths between any two points in a network. Its calculation method is as follows:

$$L(G) = \frac{1}{N(N-1)} \sum_{i,j \in V_G, i \neq j} d_{ij} \tag{6}$$

In the word network graph, the smaller the shortest path length d_{ij} between nodes i and j, the closer the relationship between the words they represent. Consequently, the higher the relevance between these two words, the more important the words connected to a key word become. Generally, when the average path length $L(i)$ of a node i representing a word is small, as the average $L(G)$ of the shortest path lengths between any two points in the network is necessarily greater than $L(i)$, the ratio of $L(i)$ to the average path length of the network $L(G)$ is less than 1. This means that the characteristics of related words must align with those nodes for which this ratio is less than 1. Such nodes are marked as significant to simplify the complexity of extracting word features later. Thus, the condition for assessing the relevance of connected nodes, L^*, is defined as follows:

$$L^* = \frac{L(i)}{L(G)}, L^* < 1 \tag{7}$$

3.2 The Comprehensive Weight of Words

In the graph-based keyword extraction methods, the weight of nodes plays a crucial role, with the accuracy of node weight assessment being pivotal for efficiently extracting keywords from text. Given a node's significance may depend on various factors, this paper combines different feature parameters to calculate the comprehensive weight of words as the node's weight.

When the word network graph G of words in a text exhibits small-world characteristics, there are words that result in the word network graph G having a shorter average path length $L(G)$ or a higher clustering coefficient $C(G)$, thus leading to a word's clustering coefficient $C(i)$ being greater than the network's clustering coefficient $C(G)$, or a word's average path length $L(i)$ being less than the network's average path length $L(G)$, meeting $C^* > 1$ or $L^* < 1$. Based on these characteristics, the WS weight calculation for words is defined as follows:

$$WS(i) = \begin{cases} C^* \times \frac{1}{L^*}, C^* > \text{ land } L^* < 1 \\ \frac{1}{C^*} \times L^*, C^* < \text{ land } L^* > 1 \\ \left(C^* \times L^* + \frac{1}{C^*} \times \frac{1}{L^*}\right)/2, \text{ else} \end{cases} \tag{8}$$

(b) Part-of-speech weight pos(i) The part of speech of a word significantly determines its potential to become a keyword. For instance, nouns are more likely to be keywords compared to adverbs. Therefore, part of speech is considered one of the feature attributes for keyword extraction in this study. The part-of-speech weight was analyzed by statistical analysis of the part-of-speech distribution of keywords in a subset of texts, as shown in Table 1.

It can be inferred from Table 1 that the five most significant parts of speech in keywords, in descending order, are nouns, verbs, adjectives, and adverbs, accounting for 96% of the total number of keywords. Therefore, the proportions listed in the table are set as the part-of-speech weight values pos(i).

(c) Position weight loc(i)

Table 1. Part-of-speech distribution

Part of Speech	Nouns	Verbs	Adjectives	Adverbs	Others
Number	7900	4190	1975	775	640
Proportion	0.511	0.271	0.127	0.051	0.040

The position of words plays a significant role in determining their importance. Generally, the probability of keywords appearing at the beginning or end of sentences is much higher than at other positions. Therefore, higher weights are assigned to keywords appearing in specific positions. Based on the characteristics of the document, this study assigns values to the importance of words first appearing at the beginning, end, and other positions of sentences:

$$loc(i)\begin{cases}\lambda, & \text{The word corresponding to } i \text{ appears at the beginning or end of the sentence.}\\ 1, & \text{other conditions.}\end{cases} \tag{9}$$

When the position weight reaches a certain value, the actual effect of the word on the document is exaggerated, thereby reducing the accuracy of keyword extraction. Thus, there is a peak accuracy value for position weight. By selecting a subset of text data for experimentation and incrementally choosing λ values between 1 and 100 for testing, the results, as shown in Table 2, are organized and plotted in Figure 2. Based on the optimal value of λ identified in Figure 1, which is 5, the position weight for words appearing at the beginning or end of sentences is set to 5, while the weight for other positions is set to 1.

Table 2. The impact of different λ values on precision

λ	1	2	3	4	5	6	7	8
Precision (%)	26.67	29.16	29.16	29.98	30.82	30.00	29.17	29.17
λ	9	10	20	40	60	80	100	
Precision (%)	29.17	29.17	28.33	28.33	27.5	26.66	25.85	

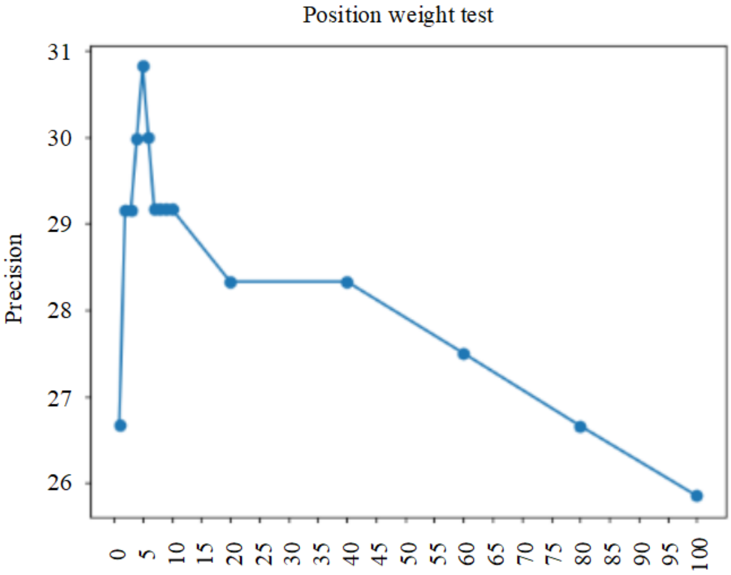


Figure 2. The impact of different λ values on precision

(d) TF-IDF weight $w_{tf}(i)$

$$w_{tf}(i) = tf_i \times idf_i \tag{10}$$

where, tf_i denotes the TF of the node, and idf_i represents the IDF of the node.

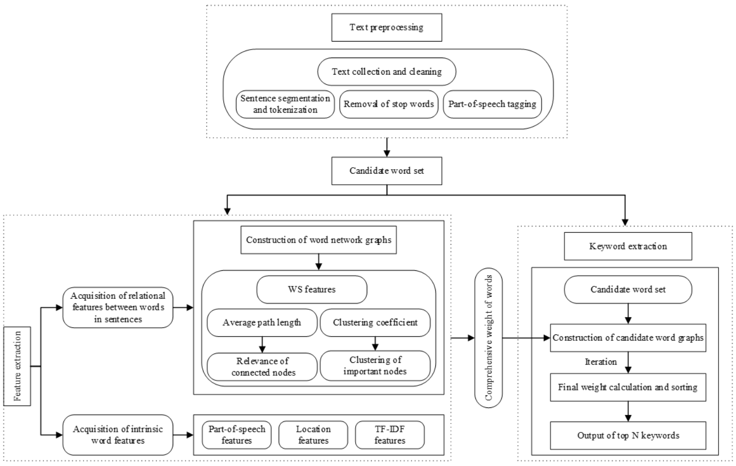
(e) Comprehensive weight $W(i)$

This study calculates the comprehensive weight of words by combining WS features, part-of-speech weight $pos(i)$, position weight $loc(i)$, and TF-IDF weight $W_{tf}(i)$, considering both the relatedness and cohesive structure between words as well as the attributes of the words themselves. The formula for calculating the comprehensive weight $W(i)$ is as follows:

$$W(i) = WS(i) \times pos(i) \times loc(i) \times w_{tf}(i) \tag{11}$$

Eq. (11) modifies the initial weight in Eq. (1) from 1 to $W(i)$, where $W(i)$ represents the comprehensive weight of words integrating WS small-world properties, and part-of-speech, positional and TF-IDF features. It considers not only the inherent characteristics of words but also leverages the Watts-Strogatz model to reflect the relevance among words in the text and the contribution of words to the overall cohesion of the text.

The algorithm proposed in this study first constructs a word network graph using the processed set of candidate words as nodes of the word graph, obtaining the WS features of the words. It then utilizes co-occurrence relationships to establish edges between nodes in an undirected graph, where the presence of two words within a co-occurrence window K indicates an edge between them, with K set to 2. Subsequently, edge weights in the undirected graph within the TextRank algorithm are set to the comprehensive weight $W(i)$ derived from the Watts-Strogatz model, part-of-speech, position, and TF-IDF, to finally calculate the weight of words and extract keywords. The workflow of the algorithm is illustrated in Figure 3.



Text preprocessing
Text collection and cleaning
Sentence segmentation and tokenization
Removal of stop words
Part-of-speech tagging
Candidate word set
Construction of word network graphs
WS features
Average path length
Clustering coefficient
Relevance of connected nodes
Clustering of important nodes
Acquisition of relational features between words in sentences
Feature extraction
Acquisition of intrinsic word features
Part-of-speech features
Location features
TF-IDF features
Comprehensive weight of words
Keyword extraction
Candidate word set
Construction of candidate word graphs
Iteration
Final weight calculation and sorting
Output of top N keywords

Figure 3. Workflow of the WS-TextRank algorithm

4 Experiment and Analysis

To evaluate the performance of the proposed algorithm, this study utilized web scraping techniques to select 1,000 academic paper abstracts from China National Knowledge Infrastructure (CNKI) to construct an assessment dataset. Comparative experiments were conducted on this dataset between the original TextRank algorithm and the improved WS-TextRank algorithm.

4.1 Data Preparation

The experiment was conducted on a Chinese dataset, with academic paper abstracts serving as an important text category. Due to their more standardized form of expression, “chemical engineering”, “biology”, “computer science”, “mathematics” and “literature” were entered as keywords in CNKI to filter out test corpora comprising 1,000 segments with complete keyword tags and coherent descriptions. The marked keywords by the authors were extracted as the annotated keywords for the corpus set. Text was processed for sentence segmentation, tokenization, removal of stop words, and part-of-speech tagging before acquiring word vectors. Full-mode tokenization using the Jieba library was employed to extract all possible words, and the resulting word sets from tokenization were tagged with parts of speech.

4.2 Comparative Experiment

The experiment assessed the performance of the improved TextRank algorithm using precision, recall, and F1-score, comparing it with the TextRank algorithm and varying the number of extracted keywords to compare the precision, recall, and F1-score of keyword extraction by the algorithms. The results are presented in Table 3 and Figures 4 – 6. Initially, it is evident from the figures that the assessment results of the TextRank algorithm are lower than those of the TF-IDF algorithm, indicating that the TextRank algorithm does not perform better than the TF-IDF algorithm. Furthermore, the precision of the three algorithms compared in this study gradually decreases with an increase in the number of extracted keywords, while recall gradually increases. The largest gap in precision among the three algorithms occurs at 9 extracted keywords, with the WS-TextRank algorithm achieving the highest F1-score, indicating optimal performance when n=9. Overall, the precision, recall, and F1-score of the WS-TextRank algorithm consistently exceed those of the TF-IDF and TextRank algorithms, demonstrating superior performance of this algorithm over the other two. This suggests that the TextRank algorithm improved by integrating Watts-Strogatz model features, along with positional, part-of-speech, and TF-IDF features, enhances the quality of keyword extraction.

Table 3. Results of keyword extraction

	Precision			**Recall**			**F1-Score**		
	TF-IDH	**TextRank**	**WS-TextRank**	**TF-IDF**	**TextRank**	**WS-TextRanl**	**TF-IDF**	**TextRank**	**WS-TextRank**
$n = 5$	0.466	0.422	0.466	0.386	0.348	0.391	0.418	0.378	0.422
$n = 7$	0.397	0.349	0.412	0.458	0.400	0.472	0.421	0.369	0.437
$n = 9$	0.357	0.296	0.382	0.531	0.440	0.561	0.424	0.351	0.451
$n = 11$	0.313	0.283	0.343	0.572	0.519	0.615	0.402	0.363	0.438
$n = 13$	0.282	0.248	0.290	0.606	0.533	0.615	0.382	0.336	0.392

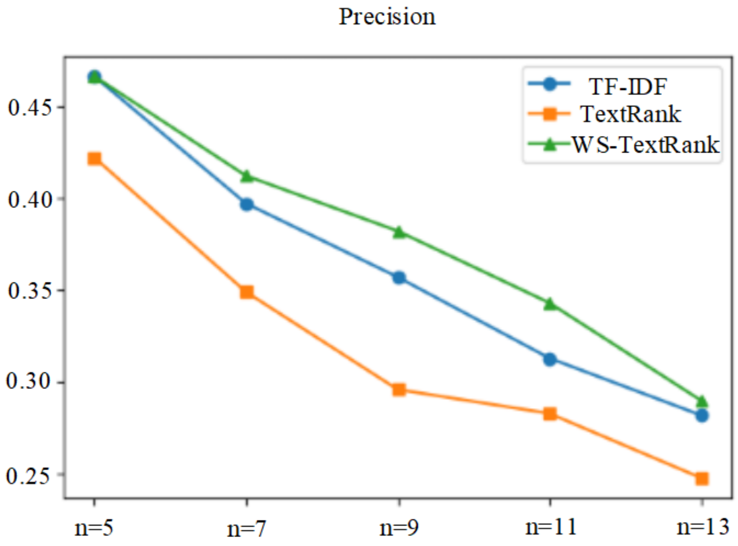


Figure 4. Precision results of keyword extraction

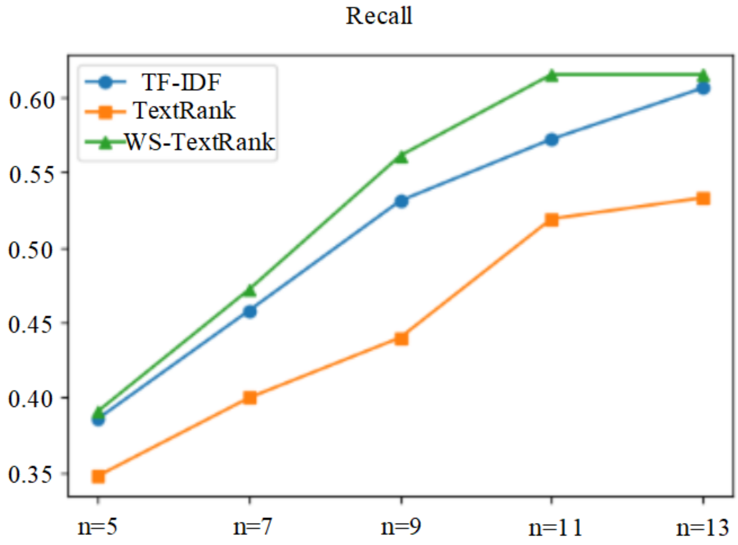


Figure 5. Recall results of keyword extraction

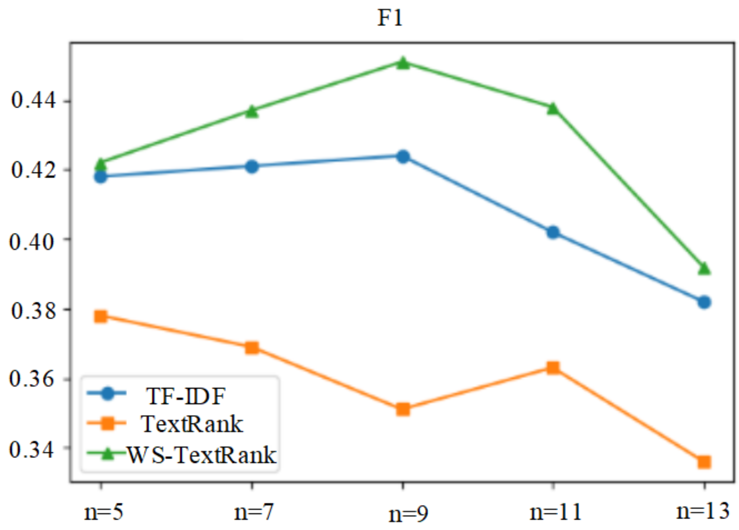


Figure 6. F1-score results of keyword extraction

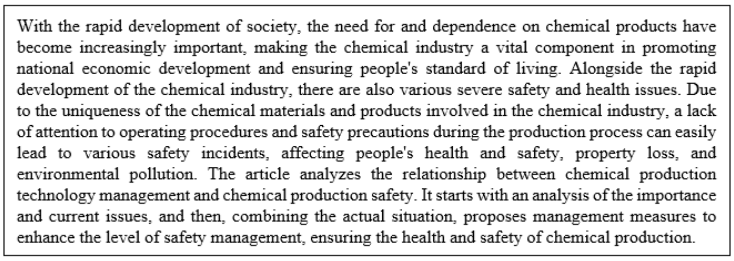
To further validate the effectiveness of the algorithm, this study obtained 500 news articles across 10 categories from the "University of Science and Technology of China Natural Language Processing and Information Retrieval Sharing Platform" (www.NLPIR.org) for experimentation. The authors had already provided keywords for these articles. Given the greater length of news articles compared to abstract texts, the experiment extracted the top 15 keywords using the original TF-IDF algorithm, the TF-IDF+TextRank algorithm, and the WS-TextRank for comparative experimentation. The comparative results, as shown in Table 4, are based on the average calculations from 50 articles. Previous experimental results indicated that the TextRank algorithm does not outperform the TF-IDF algorithm. As seen from Table 4, combining the TF-IDF and TextRank algorithms evidently surpasses the performance of the original TF-IDF algorithm. Building on this, by leveraging the Watts-Strogatz model to consider the relatedness and cohesive structure of words, and integrating word position, part-of-speech, and TF-IDF features to improve the TextRank algorithm, the proposed WS-TextRank algorithm demonstrated higher precision, recall, and F1-scores across the 10 categories of news articles than the other two algorithms, further verifying the effectiveness of the algorithm introduced in this study.

Table 4. Comparative results of news article experiment

	Precision			**Recall**			**F1-Score**		
Experimental Data (50)	**TF-IDF**	**TF-IDF+ TextRank**	**WS-TextRank**	**TF-IDF**	**TF-IDF+ TextRank**	**WS-TextRank**	**TF-IDF**	**TF-IDF+ TextRank**	**WS-TextRank**
Art	0.500	0.526	0.686	0.658	0.694	0.896	0.567	0.597	0.775
Literature	0.453	0.566	0.746	0.483	0.690	0.910	0.497	0.622	0.819
Education	0.466	0.593	0.726	0.583	0.736	0.901	0.517	0.656	0.803
Philosophy	0.433	0.586	0.733	0.520	0.706	0.880	0.472	0.640	0.799
History	0.460	0.593	0.786	0.526	0.678	0.901	0.490	0.632	0.839
Space	0.520	0.615	0.746	0.609	0.690	0.863	0.560	0.649	0.799
Energy	0.420	0.553	0.780	0.477	0.627	0.886	0.446	0.588	0.829
Electronics	0.326	0.546	0.706	0.403	0.678	0.878	0.360	0.604	0.781
Communication	0.466	0.606	0.766	0.496	0.671	0.853	0.469	0.636	0.806
Computer	0.493	0.646	0.766	0.543	0.707	0.839	0.516	0.674	0.800

4.3 Case Study

In this experiment, the text example shown in Figure 7 was subjected to keyword extraction using the TF-IDF, TextRank, and WS-TextRank algorithms, extracting 5, 7, 9, 11, and 13 keywords, respectively, and compared with baseline keywords. The results, as presented in Table 5, where $count_{right}$ is the number of correctly extracted keywords. Table 4 clearly demonstrates that the WS-TextRank algorithm, by considering features such as TF, position, part of speech, and the relatedness and cohesive structure of the word set, compensates for the deficiencies of the original TextRank in extracting text keywords, resulting in the highest number of keywords extracted. The analysis of the sample instance extraction results from this experiment further validates the superiority of the WS-TextRank algorithm in extracting text keywords.



With the rapid development of society, the need for and dependence on chemical products have become increasingly important, making the chemical industry a vital component in promoting national economic development and ensuring people's standard of living. Alongside the rapid development of the chemical industry, there are also various severe safety and health issues. Due to the uniqueness of the chemical materials and products involved in the chemical industry, a lack of attention to operating procedures and safety precautions during the production process can easily lead to various safety incidents, affecting people's health and safety, property loss, and environmental pollution. The article analyzes the relationship between chemical production technology management and chemical production safety. It starts with an analysis of the importance and current issues, and then, combining the actual situation, proposes management measures to enhance the level of safety management, ensuring the health and safety of chemical production.

Figure 7. Example text

Table 5. Comparative results of different top N keyword extraction instances

Top N	Algorithm	Keywords	$count_{right}$
5	TF-IDF	‘chemical engineering’, ‘park’, ‘high quality’, ‘platform’, ‘introduction’	3
	TextRank	‘park’, ‘chemical engineering’, ‘high quality’, ‘platform’, ‘group’	3
	WS-TextRank	‘park’, ‘chemical engineering’, ‘practice’, ‘platform’, ‘high quality’	4
7	TF-IDF	‘chemical engineering’, ‘park’, ‘high quality’, ‘platform’, ‘introduction’, ‘practice’, ‘prevention’	4
	TextRank	‘park’, ‘chemical engineering’, ‘practice’, ‘platform’, ‘high quality’, ‘introduction’, ‘practical’	4
	WS-TextRank	‘park’, ‘chemical engineering’, ‘high quality’, ‘platform’, ‘group’, ‘digitalization’, ‘special duty’	5
9	TF-IDF	‘chemical engineering’, ‘park’, ‘high quality’, ‘platform’, ‘introduction’, ‘practice’, ‘prevention’, ‘chemical enterprises’, ‘risk’	4
	TextRank	‘park’, ‘chemical engineering’, ‘practice’, ‘platform’, ‘high quality’, ‘introduction’, ‘practical’, ‘team’, ‘weakness’	4
	WS-TextRank	‘park’, ‘chemical engineering’, ‘high quality’, ‘platform’, ‘group’, ‘digitalization’, ‘special duty’, ‘fine chemical engineering’, ‘practice’	6
11	TF-IDF	‘chemical engineering’, ‘park’, ‘high quality’, ‘platform’, ‘introduction’, ‘practice’, ‘prevention’, ‘chemical enterprises’, ‘risk’, ‘main battlefield’, ‘industry’	4
	TextRank	‘park’, ‘chemical engineering’, ‘practice’, ‘platform’, ‘high quality’, ‘introduction’, ‘practical’, ‘team’, ‘weakness’, ‘fine chemical engineering’, ‘Crane Mountain’	4
	WS-TextRank	‘park’, ‘chemical engineering’, ‘high quality’, ‘platform’, ‘group’, ‘digitalization’, ‘special duty’, ‘fine chemical engineering’, ‘practice’, ‘risk’, ‘carrier’	7
13	TF-IDF	‘chemical engineering’, ‘park’, ‘high quality’, ‘platform’, ‘introduction’, ‘practice’, ‘prevention’, ‘chemical enterprises’, ‘risk’, ‘main battlefield’, ‘industry’, ‘carrier’, ‘resolution’	5
	TextRank	‘park’, ‘chemical engineering’, ‘practice’, ‘platform’, ‘high quality’, ‘introduction’, ‘practical’, ‘team’, ‘weakness’, ‘fine chemical engineering’, ‘Crane Mountain’, ‘management center’, ‘industrial park’	4
	WS-TextRank	‘park’, ‘chemical engineering’, ‘high quality’, ‘platform’, ‘group’, ‘digitalization’, ‘special duty’, ‘fine chemical engineering’, ‘practice’, ‘risk’, ‘carrier’, ‘specialization’, ‘economy’	7

5 Conclusion

To address the inability of the TextRank algorithm to capture the contextual complexity of documents and its lack of consideration for the relatedness and cohesive structure between sets of words in Chinese text keyword extraction, this study proposes an improved TextRank keyword extraction method based on the Watts-Strogatz model to enhance the effectiveness of keyword extraction. The method integrates the basic concepts of the Watts-Strogatz model, utilizing the characteristics of short average path lengths and high clustering coefficients of SWNs. It introduces evaluation conditions for node relevance and clustering to obtain the WS features of words. These features, combined with the part-of-speech, position, and TF-IDF features of words, are used to calculate the comprehensive weight of words. This weight is then applied to modify the influencing factors of the TextRank algorithm, determining the final weight of candidate words in the text. Ranking the candidate words based on their final weight identifies the top N words as keywords. Experiments conducted on a Chinese academic abstract text dataset demonstrate that the method is feasible and effective for extracting keywords from Chinese texts, yielding superior results. However, the method requires constructing two graphs for the word set, with one being the construction of small-world word networks when extracting WS features, and another being the word graph construction before executing the TextRank algorithm, which leads to relatively high time complexity. Therefore, future work will focus on reducing the time complexity of keyword extraction while ensuring accuracy, making the algorithm more efficient.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Nadim, D. Akopian, and A. Matamoros, “A comparative assessment of unsupervised keyword extraction tools,” *IEEE Access*, 2023. https://doi.org/10.1109/ACCESS.2023.3344032

[2] A. Vanyushkin and L. Graschenko, “Analysis of text collections for the purposes of keyword extraction task,” *J. Inf. Organ. Sci.*, vol. 44, no. 1, pp. 171–184, 2020. https://doi.org/10.31341/jios.44.1.8

[3] H. P. Luhn, “A statistical approach to mechanized encoding and searching of literary information,” *IBM J. Res. Dev.*, vol. 1, no. 4, pp. 309–317, 1957. https://doi.org/10.1147/rd.14.0309

[4] B. M. Jafari, X. Luo, and A. Jafari, “Unsupervised keyword extraction for hashtag recommendation in social media,” in *the International FLAIRS Conference Proceedings*, 2023. http://doi.org/10.32473/flairs.36.133280

[5] T. Zhang, B. Lee, Q. Zhu, X. Han, and K. Chen, “Document keyword extraction based on semantic hierarchical graph model,” *Scientometrics*, vol. 128, no. 5, pp. 2623–2647, 2023. https://doi.org/10.1007/s11192-023-04677-7

[6] R. Devika and V. Subramaniyaswamy, “A semantic graph-based keyword extraction model using ranking method on big social data,” *Wirel. Netw.*, vol. 27, no. 8, pp. 5447–5459, 2021. https://doi.org/10.1007/s11276-019-02128-x

[7] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, 1998. https://doi.org/10.1038/30918

[8] R. F. I. Cancho and R. V. Solé, “The small world of human language,” *Proc. R. Soc. Lond. B Biol. Sci.*, vol. 268, no. 1482, pp. 2261–2265, 2001. https://doi.org/10.1098/rspb.2001.1800

[9] L. Ma, L. Jiao, L. Bai, Y. Zhou, and L. Dong, “Research on a compound keywords detection method based on small world model,” *J. Chin. Inf. Process.*, vol. 2009, no. 3, 2009. https://doi.org/10.1109/ICEMI.2009.5274436

[10] K. Sparck Jones, “A statistical interpretation of term specificity and its application in retrieval,” *J. Doc.*, vol. 28, no. 1, pp. 11–21, 1972. https://doi.org/10.1108/eb026526

[11] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, 1988. https://doi.org/10.1016/0306-4573(88)90021-0

[12] X. Huang and W. Li, “Chinese keyword extraction method based on multi-features,” *Comput. Modern.*, vol. 2013, no. 4, pp. 15–17, 2013. https://doi.org/10.3969/j.issn.1006-2475.2013.04.004

[13] X. L. Ding and L. C. Wang, “Research on optimized calculation method for weight of terms in BBS text,” *Inf. Stud. Theory Appl.*, vol. 44, no. 5, pp. 187–192, 2021.

[14] M. Bounabi, K. Elmoutaouakil, and K. Satori, “A new neutrosophic TF-IDF term weighting for text mining tasks: Text classification use case,” *Int. J. Web Inf. Syst.*, vol. 17, no. 3, pp. 229–249, 2021. https://doi.org/10.1108/IJWIS-11-2020-0067

[15] R. Mihalcea and P. Tarau, “Textrank: Bringing order into texts,” in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain*, 2004, pp. 404–411.

[16] X. J. Wan and J. G. Xiao, "Single document keyphrase extraction using neighborhood knowledge," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, Chicago, Illinois, USA*, 2008, pp. 855–860.

[17] S. Danesh, T. Sumner, and J. H. Martin, "Sgrank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction," in *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics, Denver, Colorado, USA*, 2015, pp. 117–126.

[18] C. Florescu and C. Caragea, "Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada*, 2017, pp. 1105–1115. https://doi.org/10.18653/v1/P17-1102

[19] H. Dong, J. Wan, and Z. Wan, "Keyphrase extraction based on multi-feature," in *2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, Taiyuan, China, 2019, pp. 208–213. https://doi.org/10.1007/s40815-021-01190-y

[20] D. Qiu and Q. Zheng, "Improving textrank algorithm for automatic keyword extraction with tolerance rough set," *Int. J. Fuzzy Syst.*, vol. 24, pp. 1332–1342, 2022. https://doi.org/10.1007/s40815-021-01190-y

[21] Y. Matsuo, Y. Ohsawa, and M. Ishizuka, "Keyworld: Extracting keywords from document s small world," in *Discovery Science: 4th International Conference, DS 2001 Washington, DC, USA*, 2001, pp. 271–281. https://doi.org/10.1007/3-540-45650-3_24

[22] M. Zhang, H. T. Geng, and X. Wang, "Automatic keyword extraction algorithm research using BC method," *Minimicro Syst. Shenyang*, vol. 28, no. 1, p. 189, 2007. https://doi.org/10.1109/GLOCOM.2007.8

[23] S. Yang, Y. Liu, Y. Zhang, and J. Zhu, "A word-concept heterogeneous graph convolutional network for short text classification," *Neural Process. Lett.*, vol. 55, no. 1, pp. 735–750, 2023. https://doi.org/10.1007/s11063-022-10906-6

[24] M. X. Zhu, Z. Cai, and Q. S. Cai, "Automatic keywords extraction of Chinese document using small world structure," in *International Conference on Natural Language Processing and Knowledge Engineering (NLPKE 2003), Beijing, China*, 2003, pp. 438–443. https://doi.org/10.1109/NLPKE.2003.1275946

[25] L. B. Dong, "Research on a compound keywords abstraction based on small world network theory," Ph.D. dissertation, Xidian University, 2007.