Scientometrics (2023) 128:2623–2647
https://doi.org/10.1007/s11192-023-04677-7

Document keyword extraction based on semantic hierarchical graph model

Tingting Zhang[1] · Baozhen Lee[1] · Qinghua Zhu[2] · Xi Han[3] · Ke Chen[1]

Received: 31 January 2021 / Accepted: 24 February 2023 / Published online: 30 March 2023

Abstract

Keyword provide a brief profile of document contents and serve as an important method for quickly obtaining the document's themes. Traditional keyword extraction methods are mostly based on statistical relationships between words, with no deeper understanding of the words' structures. In addition, most studies to date performing keyword extraction are based on ranking-related measure values, without considering the cohesion of the extracted keyword set. In this paper, a keyword extraction method based on a semantic hierarchical graph model is proposed. First, the semantic graph for the document is constructed based on the hierarchical extraction of feature terms. Then, the keyword collection of the document is chosen from the constructed semantic graph. The keyword extraction method in this paper fully accounts for both the context of the keywords and the internal structure by which they are related. By mining the deep hidden structure of feature terms, the proposed method can effectively reveal the hierarchical association between terms within the semantic graph and obtain a keyword collection result with high probability. Moreover, several experiments conducted on released datasets show that our method outperforms the existing methods in terms of precision, recall, and F-measure.

Keywords Semantic hierarchical graph · Keyword extraction · Feature terms · Text mining

Introduction

Keywords are defined as single- or multiword units that capture the main topics of an underlying document (Hashemzahde et al., 2020; Onan et al., 2016; Xie et al., 2017). A document's keywords play an important role in many fields, such as information retrieval, text summarization, automatic classification, clustering, and indexing (Naidu et al., 2018; Tutkan et al., 2016; Vanyushkin et al., 2020). The manual assignment of keywords is laborious, costly, and time-consuming. Therefore, the automatic keyword

✉ Baozhen Lee
bzli@nau.edu.cn

[1] School of Information Engineering, Nanjing Audit University, Nanjing 211815, China

[2] School of Information Management, Nanjing University, Nanjing 210023, China

[3] School of Business Administration, Guangdong University of Finance and Economics, Guangzhou 510320, China

extraction problem has naturally engaged researchers' attention (Nasar et al., 2018; Qian et al., 2018).

Keyword extraction is the process of quickly obtaining important topical words or phrases from the document body, enabling a compact representation of the document's content and facilitating more comprehensive text analysis (Gopan et al., 2020; Papagiannopoulou et al., 2019). Approaches to automating this activity are usually classed as either supervised or unsupervised (Jose & Rahamathulla, 2016; Ying et al., 2017). The former requires a high labor cost. Therefore, existing keyword extraction methods focus primarily on unsupervised learning approaches that are most widely applicable and least sensitive. The unsupervised approaches proposed thus far have involved many techniques, the most prominent of which include (i) statistical methods, which focus on statistics derived from nonlinguistic features of the document, (ii) topic-based methods, which use the distribution of topics, and (iii) graph-based methods, which construct a language network graph for the document. Among these three categories, graph-based methods perform best in terms of the effective disclosure of structure and relationships within the document (Ying et al., 2017).

Typically, the graph is first constructed based on words and word relationships. Then, a word scoring method is applied according to specific measures such as centrality and frequency. Finally, the most important words are selected according to their respective scores. Most of these methods, however, base their concept of "relationship" solely on word correlations, such as co-occurrence within a fixed window size or semantic correlations derived from an external knowledge base (Beliga et al., 2015; Blanco & Lioma, 2012; Wang et al., 2020). Consequently, these methods fail to consider the impact of deep structure mining between words. This may lead to the loss of some important information about the whole document, especially when complex hierarchical relationships are involved. Moreover, the graph-based keyword extraction methods used in most research have based their keyword selection on ranking individual nodes according to a chosen importance measure (Kumar & Rehan, 2021; Ying et al., 2017). In reality, document keywords must not only be individually relevant (as established via centrality, connections, etc.) but also form cohesive structures that reflect the regular optimal combination among keywords. That is, the selected keywords must completely represent the document theme(s) in the form of a set. The discovery of a means of evaluating the internal correlation between these keywords and determining the optimal such set is, therefore, a major outstanding problem in keyword extraction research.

Given the above considerations, we propose a new keyword extraction method that exploits a richer semantic hierarchical graph. The terms in the graph are not simple words but feature terms that represent the content of the document. Two main processes are involved: (1) a semantic graph is constructed based on the hierarchical extraction of feature terms, and (2) keyword extraction is performed based on the semantic graph created. During graph construction, both similarities and distances between feature terms are considered; we depart from the established graph construction process by creating the graph hierarchically based on a predefined correlation threshold for feature terms. In the process of keyword extraction, the importance intensity of feature terms is first obtained to generate the candidate keyword graph. Then, a new measure is proposed to calculate the joint probability of the predefined number of extracted keywords derived from the candidate keyword graph. This measure is based on the probabilistic retrieval model to consider both content and structure, resulting in a set of keywords that mutually cohesively reflect the thematic features of the document.

The results of several experiments on a released data sample indicate that the proposed keyword extraction method outperforms existing methods in terms of precision, recall, and F-measure and ranking quality measures. The main advantage of our method is that it reflects the topic feature relationship and inherently hierarchical structure of the feature terms, and captures the highest-contributing collection of feature terms for the document.

The remainder of this paper is organized as follows. In Sect. "*Related work*", we offer a survey of prior work related to our research area. Section "*Graph-based method for keyword extraction*" describes our methods: the framework of the graph-based keyword extraction model is first presented, followed by an overview of our semantic hierarchical graph model based on the document's feature terms, with the remainder of the section devoted to details of our keyword extraction method as derived from the foregoing model. In Sect. "*Experimental analysis*", we evaluate the performance of the proposed method via various experiments on a real dataset; the results are discussed in Sect. "*Discussion*", and conclusions are drawn in Sect. "*Conclusion*".

Related work

Some researchers regard keyword extraction as a classification problem for which a supervised method is suitable (Chidambaram & Srinivasagan, 2016; Treeratpituk et al., 2010). Such methods determine whether a word or phrase belongs to a "keyword" or "nonkeyword" category according to a learning model built through training data. Like other supervised classification or labeling learning methods, these rely on a variety of models, including naive NB (Alqaryouti et al., 2018), SVM (Zhang, 2006), CRF (Zhang, 2008) and KEA (Witten et al., 2005). Unsupervised keyword extraction methods fall into three major categories: statistical methods, topic-based methods, and graph-based methods.

Statistical methods

These extract a document's keywords by using the statistical information associated with individual words, requiring neither training data nor an external knowledge base. A word vector is obtained after the document is preprocessed, and then a set of candidate words (phrases) is formed based on simple statistical rules such as n-gram, part-of-speech filtering, term frequency (TF) or term frequency-inverse document frequency (TF-IDF), cooccurrence, or position (Aizawa, 2003; Campos et al., 2018; Siddiqi & Sharan, 2015; Xu et al., 2021). Of these, TF-IDF has become the mainstream statistical method because of its generalizability and ease of implementation. This method calculates a word's score as the product of its TF and IDF values and selects the highest-scoring words as keywords based on the sorting scores of the words. However, TF-IDF does not consider the semantic association patterns within the document. It ignores important low-frequency words and topic distribution within the document. Another keyphrase extraction system, KP-Miner (El-Beltagy et al., 2009), can be applied in English and Arabic documents as the rules and heuristics adopted by the system are related to the general nature of documents and keyphrases.

Topic-based methods

Topic-based methods aim to extract keywords based primarily on the topic distribution of a given document. One such method, latent Dirichlet allocation (LDA) (Blei et al., 2003), works by assigning document text to a particular topic. This is an unsupervised machine learning technology that can be used to identify hidden topic information in a large-scale document corpus. Conceptually, LDA models a document as a "bag of words," which is implemented as a word frequency vector; the candidate keywords of each document can be obtained by assigning the words contained in the subject to the document. LDA extracts keywords by using the implicit document semantic information, but the keywords thus extracted are relatively broad, failing to reflect the document's theme very well.

Although topic-based models are a relatively young avenue of research, their applications have become increasingly extensive, so such models now play a large role in the automatic keyword extraction process. Pu et al. (2015) proposed a topic distilling with compressive sensing (TDCS) model, which analyzes implicit topics for document keywords using unsupervised iterative methods. Bougouin et al. (2013) proposed a keyword exaction model that relies on a topical representation of the document. It represents a document as a graph where vertices are not terms but topics composed of terms, and the between terms similarity depends on the overlapping of words. Liu et al. (2009) proposed KeyCluster to cluster similar candidate keywords using Wikipedia and co-occurrence statistics. They perform clustering by calculating the word relevance, and the keywords correspond to words near the center of the cluster. Another method presented by Liu et al. (2010), called topical PageRank (TPR), acquires the topics of words and documents according to LDA and then combines this information with TextRank to construct the word graph. To avoid the large cost of topical PageRank by running only one PageRank for each document, Sterckx et al. (2015) incorporated topical information from topic models. to improve the topical PageRank.

Graph-based methods

We review the graph-based methods in greater depth because these are the methods most closely related to the present study. Below, we summarize some of the main results from work in this area over the past fifteen years.

TextRank is a keyword extraction method based on a graph model that remains highly representative of the graph-based approach (Mihalcea & Tarau, 2004). It does not need to be trained on multiple documents in advance and has been widely used because of its simplicity and effectiveness. As the name might suggest, TextRank derives from PageRank (Brin & Page, 1998); it achieves keyword extraction within a single document by dividing the document into several units and building a graph model based on word connections, with important units in the document sorted by a voting mechanism. In TextRank, word connections are often computed via word co-occurrences within the document. Boudin (2018) proposed an unsupervised keyphrase extraction model that encodes topical information within a multipartite graph structure. It represents keyphrase candidates and topics in a single graph and exploits their mutually reinforcing relationship to improve candidate ranking.

Many recent studies in keyword extraction have been motivated by the concept of semantic graphs. Biswas et al. (2018) proposed an automatic keyword extraction method for Twitter in which the graph vertices are created from the set of tokens, and the edges are established by pairs of tokens occurring in the same sequence. The weight of each edge is based on the frequency of the nodes and their co-occurrence frequency. From the graph, keyword importance is measured via parameters such as frequency, centrality, position, and strength of the candidate keyword's neighbors. Tixier et al. (2016) assumed that keywords are more likely to be determined by influential nodes of a word graph that may not have many important connections rather than by eigenvector-related centrality measures. An unsupervised keyword extraction method (GoW) is introduced that capitalizes on graph degeneracy, considering the density and cohesiveness of groups of nodes. The GoW representation regards a piece of text as an undirected graph, where nodes are unique nouns and adjectives, and edges represent co-occurrence within a window of a predetermined size. To extract the keywords, the proposed CoreRank method converts the cohesiveness information captured by degeneracy into ranks and selects the top p% nodes in order of score.

Abilhoa and De Castro (2014) proposed a keyword extraction method called TKG (for "Twitter keyword graph") that represents texts as graphs and applies centrality measures to find the relevant keywords. In their graph model, one vertex is created per token, and the weights of the undirected edges are assigned according to co-occurrence, as decided by either nearest-neighbor edging or all-neighbors edging. The centrality measures include the three aspects of degree centrality, closeness centrality, and eccentricity. As above, the highest-ranked n% of vertices are selected as keywords. In the work of Rose et al. (2010), the graph of word co-occurrences is complete once every candidate keyword is recognized after the application of phrase delimiters and stop word positions. Their suggested metrics for calculating keyword scores include word frequency, word degree, and ratio of the degree to frequency. Beliga et al. (2017) introduce a selectivity-based keyword extraction method (SBKE), whose functionality is demonstrated in both Serbian and English corpora. The method consists of two phases—keyword extraction and keyword expansion—and utilizes the structural and statistical properties of text represented as a complex network. The average weight of ingoing and outgoing links to a single node is calculated from the network, and this average is then ranked for use in keyword determination. Litvake et al. (2011) proposed DegExt, a graph-based cross-lingual keyphrase extractor method. The method focuses on the simple graph-based syntactic representation of a document; node filtering is specified based on the absolute number of nodes or ratio threshold. Blanco and Lioma (2012) proposed a principled graph-theoretic method for term weighting. Two kinds of graphs are built: an undirected graph for term co-occurrence and a directed graph for term co-occurrence and grammatical dependence. From these, ranking calculations are performed by considering the topological properties of the whole graph to measure term weights. Graph-based methods treat the document as a network of words, but to the extent that these methods depend on co-occurrence or grammatical relationships, they still lack a deeper picture of the correlations and structure between words; thus, they fail to capture the contextual complexity of the document. Therefore, a graph representation with a more sophisticated form of correlation is worth exploring.

In addition to word relationships, other factors, such as themes and sentences, have been considered for use in graph-based keyword extraction. Ravinuthala and Ch (2016) proposed a thematic text graph in which weighted edges are drawn between words according to the document theme. The keyword weight is computed based on the existing centrality measure. Rafiei-Asl and Nickabadi (2017) combined theme and structure in a keyphrase extractor method based on a co-occurrence graph in combination with prior knowledge about the input language. Researchers who consider relationships at the sentence level assume that a word must be important if it is connected to other important words and appears in many important sentences. Their methods involve graphs that represent three kinds of relationships: word to word, sentence to sentence, and sentence to word. In the research of Ying et al. (2017), term clustering was performed for keyphrase selection by considering the importance of both words and sentences. Words near the centroid of each cluster were then selected as keyphrases according to the importance scores. Yang et al. (2018) assigned nodes a synthetic eigenvalue by combining contributions from a word and a sentence network. The parameterless graph construction method of Duari and Bhatnagar (2019) is based on the pragmatics of written communication and emphasizes candidate co-occurrences in which the co-occurrence window passes over two consecutive sentences. The score of a candidate keyword is defined by the level of embeddedness, semantic strength, extent of conceptual linkage, and positional weight. Figueroa et al. (2018) proposed a method called RankUp that enhances the popular graph-based keyphrase extraction methods TextRank and RAKE. An error-feedback algorithm utilizing TF-IDF, RIDF, and clusteredness was proposed that plays a role conceptually similarly to backpropagation.

Other methods

The method proposed by Kumar and Rehan (2021) is intended for real-time detection of Twitter keywords. It employs a directed weighted graph in which nodes are the words constituting an individual tweet and edges are the precedence relationships of words. The candidates are generated via the Levenshtein distance and the double metaphone algorithm in combination with a dictionary approach. Then, a candidate scorer and candidate selector are employed to select the best possible normalized word. Hulth et al. (2006) presented a study on whether and how automatically extracted keywords can be used to improve text categorization. It showed a higher performance when the full-text representation was combined with the automatically extracted keywords. Nguyen et al. (2007) presented a keyphrase extraction algorithm for scientific publications by extending the additional features that capture the logical position and additional morphological characteristics. Bougouin et al. (2013) proposed a keyword exaction model that relies on a document topical representation. It represents a document as a graph where vertices are not terms but topics composed of terms, and the similarity between terms depends on the overlapping of words. Mothe et al. (2018) integrated a word embedding representation into their keyphrase extraction process but concluded that it yields no improvement in results. Wang et al. (2015) proposed an approach that uses word embedding vectors as an external knowledge base for both keyword extraction and generation, which also shows a better performance compared with many baseline algorithms. Mahata et al. (2018) presented an unsupervised technique that leverages phrase embeddings for ranking keyphrases extracted from scientific articles by using the popular keyphrase extraction frameworks of candidate selection, scoring, and ranking.

Graph-based method for keyword extraction

In this section, we seek to effectively derive a keyword collection for a given document by mining the deep structure that exists between feature terms and combining intrinsic term association information. Computationally, we can transform the keyword extraction problem into the problem of calculating the joint probability of a specific feature term set based

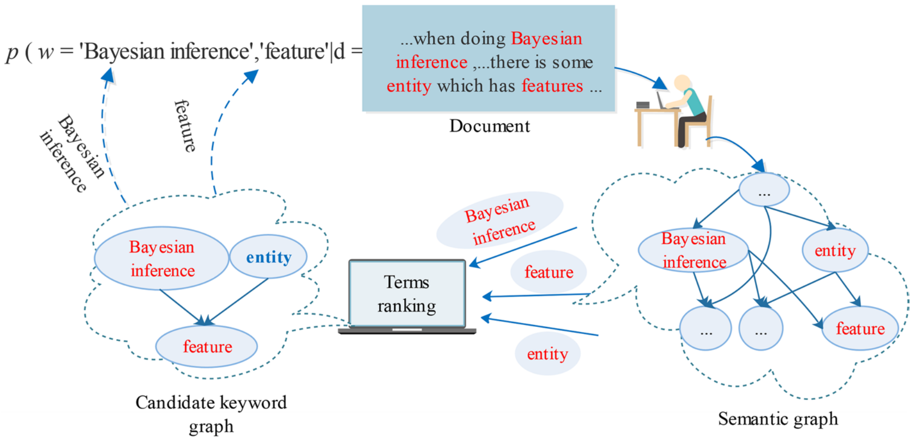

Fig. 1 Computing the probability of the given keywords based on a semantic graph

on the document. Effective keyword extraction depends not only on the feature term content but also on the deep structure of those terms.

As shown in Fig. 1, feature terms are first extracted from documents; then, the semantic graph of these terms is constructed. Next, the feature terms are ranked in descending order of importance intensity, according to the terms' content and structure information of feature terms, and this ranking is used to identify candidate keyword graphs. Finally, the optimal keyword set is selected from among the candidate keywords using the joint probability method.

As illustrated in Fig. 2, the proposed framework consists of five main components: (1) preprocessing: for document text segmentation; (2) term correlation definition: to identify correlated terms based on similarity and distance analysis; (3) semantic graph construction: to form a hierarchical graph based on term correlation; (4) candidate keyword selection: to shortlist the keywords according to node and edge information of the semantic graph; and (5) keyword selection: to retrieve the optimal candidate keywords by using the probabilistic retrieval model. Further details are elaborated in the subsequent sections.

Preprocessing

The preprocessing stage includes converting the document to a suitable data structure, segmenting the text into feature terms, and removing stop words.

Document conversion

The entire document to be analyzed is converted to paragraphs based on its physical structure since we use paragraphs as the unit for analysis. Each paragraph derived from the document is tagged to facilitate later identification.

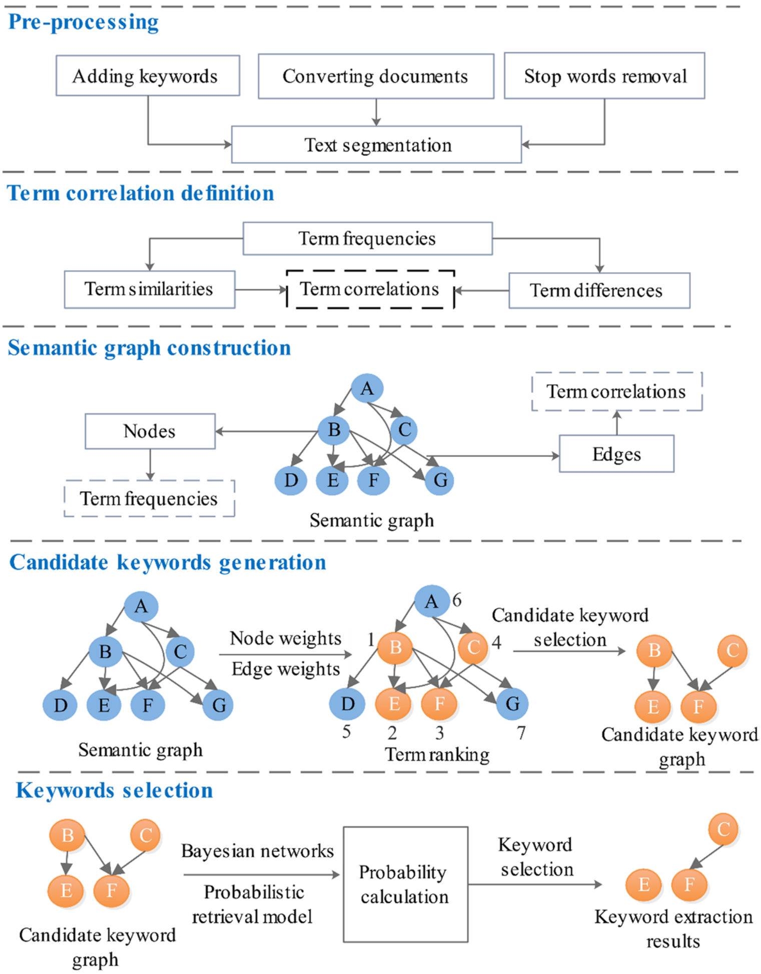

Fig. 2 Keyword extraction framework overview

Stop word removal

Stop words that do not hold any value for understanding the content of the document are removed from the set of feature terms. A list of stop words in Chinese and English, provided by GitHub,[1] is used in this research.

Text segmentation

Text segmentation entails dividing the document into a set of feature terms. The feature terms in our graph are not words in the ordinary sense but a word or phrase used to

[1] https://github.com/ImportMe/stop_words.

Table 1 The example of text segmentation

Language	Sentence	Feature terms	Stop word
Chinese	关键词提取的方法与代码	关键词、提取、方法、代码	的, 与
English	Bayesian inference is a method	Bayesian inference、method	is, a

describe a thing or to express a concept. Jieba[2] is a popular Python package for segmenting Chinese text into meaningful feature terms. Note that the terms output in this stage describe the semantic information within the document and are regarded as the basic units in this paper. For example, Table 1 shows that the Chinese sentence “关键词提取的方法与代码” can be decomposed into the feature terms “关键词, 提取, 方法, 代码”. This represents the basic unit that expresses the text content characteristics; for the same reason, the English sentence “Bayesian inference is a method” can be decomposed into “Bayesian inference, method”.

Term correlation definition

Feature terms and their structural relationships have important roles in document analysis. The stronger the structural relationships by which the terms are associated, the greater the contribution of the terms to the document.

The correlations between terms are represented by similarities (s) and differences ($1 - s$) via the distribution of term frequencies in each paragraph. The value of s is calculated based on cosine similarity. In the formula below, $r(k_\mathrm{i}, k_\mathrm{j})$ represents the correlation between term k_i and term k_j:

$$r(k_i,k_j)=\overset{\overset{s}{\downarrow}}{\frac{\sum_{p=1}^{m}(k_i^p\times k_j^p)}{\sqrt{\sum_{p=1}^{m}(k_i^p)^2}\times\sqrt{\sum_{p=1}^{m}(k_j^p)^2}}}\bullet\overset{\overset{1-s}{\downarrow}}{1-\frac{\sum_{p=1}^{m}(k_i^p\times k_j^p)}{\sqrt{\sum_{p=1}^{m}(k_i^p)^2}\times\sqrt{\sum_{p=1}^{m}(k_j^p)^2}}}\bullet \tag{1}$$

and m is the number of paragraphs in the document, while k_i^p and k_i^p indicate the frequency of term k_i and term k_j, and appearing in each paragraph. A higher value of $r(k_i, k_j)$ (abbreviated as r) indicates a stronger relationship between k_i and k_j, whereas a lower value represents a relatively weaker relationship.

Term correlations within the paper thus measure the coupling relationships between feature terms across the entire document. Cosine similarity s is a measure of similarity between two nonzero vectors of an inner product space. In the context of the same set of vector space dimensions, there are two kinds of semantic relations between two terms: (1) semantic similarity, whose cosine similarity is $s = 1$ when they are completely similar; and (2) semantic complementarity, whose cosine similarity is $s = 0$ when they are completely complimentary. When two terms appear together in the same context, the absolute semantic similarity will cause redundancy, and the absolute semantic complementarity will form

[2] https://pypi.org/project/jieba/

a contradiction. Therefore, the semantic relevance between two terms in the same context considers the above two situations, namely, r = s*(s − 1). Neither semantic redundancy nor semantic contradiction can be avoided. The revealed term correlations is revealed based on the coupled co-occurrence relationship between these topic feature terms in different paragraphs, which can effectively distinguish the topic relationship between different paragraphs.

Semantic graph construction

Document representation is an essential step for text mining tasks. Graph-based representations in particular are a powerful means of representing documents and, as such, have begun to receive more attention. Representing the document as a graph allows the retention of some important information, such as semantic relationships and internal structures. In this paper, we capitalize on a semantic graph to represent the hierarchical relationships among document terms. Nodes in our graph represent terms, and node weights are allocated using the aforementioned $r(k_1, k_2)$ values. Directed edges represent the relationships between term pairs: the direction records the sequential extraction of term information, and weight is allocated to reflect the extent of the relationship. It is worth noting that unlike the existing graph construction process, in which the term correlation is defined after the graph is created, we construct the graph hierarchically based on the predefined correlation of feature terms according to snowball sampling.

More specifically, the following work is undertaken in semantic graph construction:

Selection of root node ***k****$_i$*

The root node k_i is regarded as the first target node—the starting point for feature extraction in the graph construction—and is selected based on the maximum information gain of the feature term. The root node selection of the graph can be based on decision tree theory. Based on the information gain model of decision tree theory, we choose the feature term with the maximum information gain value that can distinguish different child nodes more effectively.

In text classification, the commonly used calculation formula for information gain is:

$$\begin{aligned} IG(t) =& H(C) - H(C/t) \\ =& \sum_{c\in C}\left[P(c,t)\log\left(\frac{P(c,t)}{P(c)P(t)}\right)\right] + \sum_{c\in C}\left[P(c,\bar{t})\log\left(\frac{P(c,\bar{t})}{P(c)P(\bar{t})}\right)\right] \end{aligned} \tag{2}$$

where t represents the information gain of the feature word, c represents the class variable, C represents the text set. $IG(t)$ represents the difference between the original entropy of the system and the conditional entropy after fixing the feature t.

It should be noted that for a specific document, if the term extraction results remain unchanged, i.e. the sampling depth is large enough, almost all the topic terms and relationships in the document, then the semantic scene is fixed and the root node is determined according to the information entropy, so the graph obtained by snowball sampling is basically stable; if the setting of sampling depth is only enough to extract part of the content of the document, then the root node maybe uncertain according to the information entropy. This paper mainly assumes that all terms of a specific document can be obtained, and the graph construction is assumed to be in certain semantic scenarios, so the root node stay the

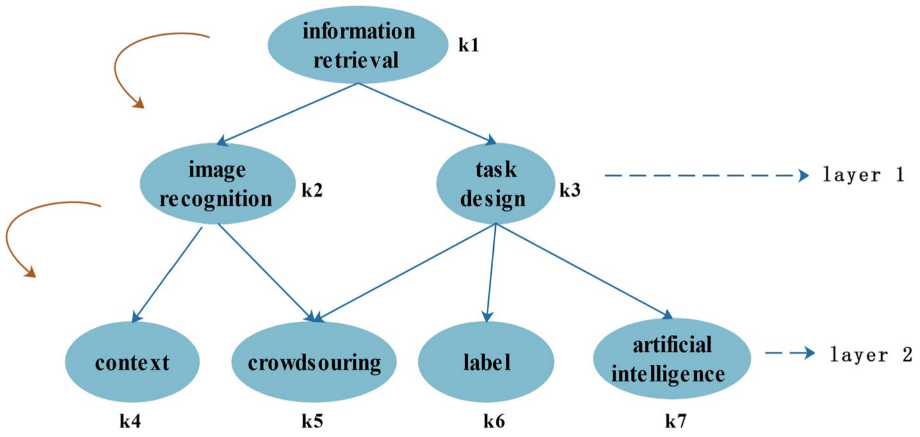

Fig. 3 Example of a semantic graph

same in our method. In addition, the root node can also be selected manually based on the user decision objectives according to task-specific analysis in different practical studies.

Selection of weight threshold *w*

The weight threshold defines the minimum weight for an edge to exist between two nodes in a graph. It is determined considering the pairwise correlations $r(k_i, k_j)$ between terms. If r between two terms is below the weight threshold, no relationship will be identified between the corresponding two nodes in the graph.

Graph model construction

Based on the parameters set above, the graph is constructed using hierarchical term extraction and snowball sampling. First, the root node k_i is selected as the target node, and the terms that have a correlation r with the target node are acquired. Then, based on w, terms whose correlations with target node k_i exceed w are selected as the first-layer nodes in the graph, described as set $\{K_1|K_1 \in \text{terms}\}$. Next, the first-layer nodes $\{K_1\}$ are designated as target nodes. Traversing $\{K_1\}$ one term at a time, the terms whose r with $\{K_1\}$ are greater than w are selected as second-layer nodes, described as set $\{K_2\ |K_2 \in \text{terms}\ \&\ K_2 \cap K_1 = \Phi\}$. Subsequent layers are selected via the same method until all terms are traversed. Finally, a hierarchical graph is assembled from the terms in their respective layers. Nodes in this graph may have multiple incoming and outgoing edges, and low-layer nodes always point to high-layer nodes. These different layers can express the document's content at different levels of granularity.

G(d) can thus be represented as a directed hierarchical graph with node set *nod.*, edge set *edg.*, node frequency N and edge weight E.

$$G(d) = G(nod., edg., N, E)$$

where *nod.* also denotes terms, *edg.* indicates term relationships, N represents term frequencies calculated based on the number of terms in the whole document, and E shows relationships between the terms as determined by term correlations r.

The graph model presented above simultaneously considers the term semantics and the structural relationship between terms. The graph thus takes a radioactive tree shape, a type of directed rooted tree also known as an antiarborescence. Nodes can be extracted repeatedly from such a tree to determine their relationships. As shown in Fig. 3, starting from the root node "information retrieval", the semantic graph is generated based on top-down node extraction by the snowball sampling technique to effectively reveal the hierarchical association between terms within the document.

As shown in Fig. 3, starting with the target root node k1, nodes k2 and k3 with high correlation with k1 are extracted as node layer 1; then, nodes k2 and k3 are regarded as the new target nodes, nodes k4 and k5 with high correlation with k2 are extracted as node layer 2, and so on; after traversing all the nodes, we obtain a final semantic graph. There is a corresponding conditional dependency between the nodes of any two layers, and the low-layer nodes always point to the high-layer nodes.

Based on the semantic hierarchy graph constructed above, the hierarchical relationship between the terms indicates a dependence association formed based on the extraction sequence choice. This dependence expresses the probability of occurrence of the next term when giving an existing term in a specific context. Additionally, it can be seen that the feature terms at different layers reflect content information at different levels of granularity, and the deep structural relationship between terms reflects the theme distribution and structural distribution of the different paragraphs.

Candidate keyword generation

In this section, we choose the nodes with higher weight values from the semantic graph as the candidate keywords. The node importance intensity in the semantic graph is calculated first, and node ranking is then employed to generate the candidate keyword graph. Then, the first M feature terms are selected to form a candidate keyword graph.

The node importance intensity (w_k) calculation depends on a combination of frequency and average adjacent edge weight:

$$w_k = \lambda \cdot \frac{N_k}{N_{nod.}} + (1-\lambda) \cdot \frac{\sum_{i=1}^{Q} E_{(k,k_i)}}{E_{edg.}} \tag{3}$$

where λ denotes a weight regulation parameter, N_k is the frequency of node k, $N_{nod.}$ is the sum of frequencies of all nodes, Q is the number of edges connecting to node k, $E_{(k,k_i)}$ represents the sum of all edge weights connected to k, and $E_{edg.}$ represents the sum of weights of all edges in $G(d)$. Terms are sorted in descending order by w_k, and redundancy elimination is performed via synonym merging. Then, the first M feature terms are selected to form the candidate keyword graph. The node set of the candidate graph is expressed as $K_{CM}=\{k_1, k_2, \ldots, k_M\}$. If we predefine the number of extracted keywords as Z ($M>Z$), then we select Z feature terms with the largest joint probability from the M candidate keywords as the document keyword extraction results.

The candidate keyword graph constructed in this study can filter out the important feature terms representing the main document content and structure. In addition, an extraction method based on term importance can greatly reduce the cost of keyword extraction. In the next section, the keywords that express the core theme of the content are finally extracted based on the candidate graph.

Keyword extraction

Document keyword collection is now selected based on the candidate keyword graph. We assume that keywords are found not among the nodes highest in individual measures of relevance (e.g., centrality) but in the optimal collection of terms from among the combinations of candidate keywords. Namely, we select the node set with maximum probability contribution value from the candidate keyword graph as the optimal keyword extraction collection.

The probabilistic retrieval model is one of the most well-known ranking models in the field of information retrieval. It is based on analyzing existing feedback results, with the current query sorted according to the Bayesian principles. Probabilistic models define their ranking function based on the probability that a given document d is relevant. In query likelihood, this probability of relevance can be approximated by the probability of a query q given a document d and relevance R, $p(q|d, R=1)$. Assume that a query q contains the words $q=\{w_1, w_2, ..., w_n\}$. The scoring or ranking function is then the probability that q is observed given that a user is thinking of a particular document d. The product of probabilities of all individual words is shown as follows, based on the independence assumption:

$$p(q|\mathrm{d}) = p\left(w_1|d\right) \times p\left(w_2|d\right) \times \ldots \times p\left(w_\mathrm{n}|d\right) \tag{4}$$

In practice, the documents available for such a query are scored using a logarithm of the query likelihood, and $p(q|d)$ is logarithmically transformed to avoid having numerous small probabilities multiplied together, which can cause underflow and precision loss.

$$Score\,(q,\, d) = \log p(q|d) = \sum_{i=1}^{n} \log p(w_i|d) \tag{5}$$

Based on the nodes in the candidate keyword graph and a number Z of pre-extracted keywords, we can obtain the feature terms with number Z from the candidate keyword graph. In this part of the workflow, we select the node set with maximum joint probability K_{set} from K_{CM} as the optimal keyword collection; this also serves as the result of keyword extraction.

For a given set of candidate keywords K_{CM}, the joint probability of Z preselected terms from K_{CM} is calculated as follows:

$$\log p\left(K_{set}|d\right) = \sum_{k_i \in \{K_{set}\}} c(k_i) \times \log\left[a_d \cdot p(E_{Kset}|G\right] \tag{6}$$

where $K_{set}=\{k_1, k_2, \ldots, k_z\}$ is the collection of Z preselected feature terms from candidate keywords. $p(K_{set}|d)$ represents the probability contribution value of K_{set} for a specific document d. $c(k)$ is the sum of the feature term frequencies—just as in the vector space model, which measures the sum of candidate feature term frequencies in G, and $p(E_{Kset}|G)$ achieves the joint weighting effect via the semantic graph, with the adjustable coefficient $a_d \mid a_d \in [0, +\infty)$ controlling the probability mass assigned to $p(E_{Kset}|G)$. where $p(E_{Kset}|G) = p(K_{k1} \mid G) \times p(K_{k2} \mid G) \times \ldots \times p(K_{kZ}| \, G)$, which is related to the product of probabilities for the feature terms' appearance in the graph. At length, the K_{set} with Z preselected terms which has maximum probability contribution value is selected as the keyword extraction result.

The proposed method in this paper considers the semantic content of the feature terms and the hierarchical relationships implied between terms. The semantic graph model based

Table 2 Overview of 9 categories in the dataset

Group no	Categories	No. of papers	Ave No. of keywords
Group 1	Art	400	7
Group 2	History	246	8
Group 3	Space	447	4
Group 4	Computer	659	4
Group 5	Environment	752	4
Group 6	Agriculture	653	6
Group 7	Economy	1102	6
Group 8	Politics	420	8
Group 9	Sports	684	6
Mean	–	–	6

on the term conditional dependency constructed in this paper is based on the correlation relationship between these topic feature terms in different paragraphs. Additionally, feature term extraction and its conditional dependence based on different paragraphs in the semantic graph can not only distinguish the topic relationship between different paragraphs but also reveal the structural topic association relationship between different paragraphs.

Experimental analysis

The objective of our proposed method is to improve the document keyword extraction quality. To evaluate the effectiveness of our method and its performance, we performed a series of experimental analyses characterized by varying parameters and different document fields, along with statistical analyses of the experimental results. Experimentation was carried out via a Windows-based program developed from a combination of Python and MATLAB.

Keyword extraction datasets

The data source we employed was from the natural language processing and information retrieval sharing platform of Fudan University in China and is publicly released (http://www.nlpir.org/wordpress/category/corpus语料库/). The platform shares many corpora that are famous for natural language processing and text analysis. The corpus contains 20 categories and more than 20,000 papers, which can be used for document topic extraction and analysis.

In the experiments, we selected 9 categories from the corpus and a subset of the documents within each category to cover different domains. In all, 5363 document items are deployed for performance evaluation. The number of documents in the chosen 9 categories is shown in Table 2.

The evaluation metrics used were precision, recall, F-measure, mean reciprocal rank (MRR), and mean average precision (MAP) (Papagiannopoulou et al. 2019). To evaluate the performance of our keyword extraction method, we compared the set of keywords generated by our method with the set of defined keywords.

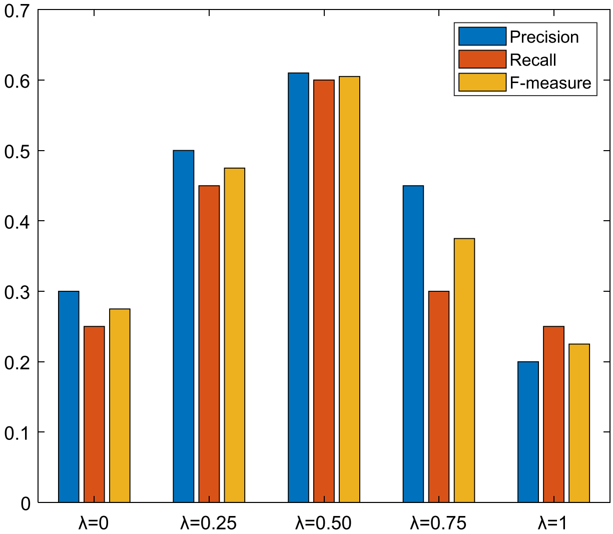

Fig. 4 Performance evaluation in terms of different λ parameters

The keyword extraction method employed in this paper uses the following parameters:

The weight threshold w (minimum value of the weight between two nodes) was set to 0.05 to extract as many high-frequency feature terms as possible. ad is the interpolation coefficient applied to the probability retrieval model and is set to document length, and λ, the weight allocation of node content in the candidate keyword graph, is set to 0.5 based on experimental results in "*variations in weighting*". The sampling depth h was set to 4 in this paper since the nodes in constructed graphs with 4 layers have captured the main content of the article.

Variations in weighting

Based on the keyword generation method used here, the importance of nodes in a candidate graph is affected by both their content and their structure. To analyze the influence of different weight parameters on the keyword results, we based our analyses on different parameter combinations. The content weight parameter λ was set to 0, 0.25, 0.5, 0.75, and 1; correspondingly, the edge weight parameter (1 – λ) was set to 1, 0.75, 0.5, 0.25, 0. The number of keywords extracted in this experiment was set to the author-defined keyword count.

We used precision, recall, and F-measure as experimental criteria. First, we calculated the precision, recall, and F-measure of every evaluation dataset. Then, we calculated the average precision, recall, and F-measure of all evaluation datasets. The experimental results are as follows.

Figure 4 shows the precision, recall, and F-measure under different λ parameters. It can be seen from the result that all three evaluation metrics exceed 0.6 when $\lambda = 0.5$. The metrics display lower values (<0.3) at the extremes ($\lambda = 0$ and $\lambda = 1$). According to this analysis, the parameter value choice has a definite impact on the keyword extraction results. When $\lambda = 0$, only the relationships between feature terms are considered, and the influence of the content is ignored. Conversely, when $\lambda = 1$, only the frequency of feature terms is considered, and the influence of the structure between terms is ignored. Therefore, we set

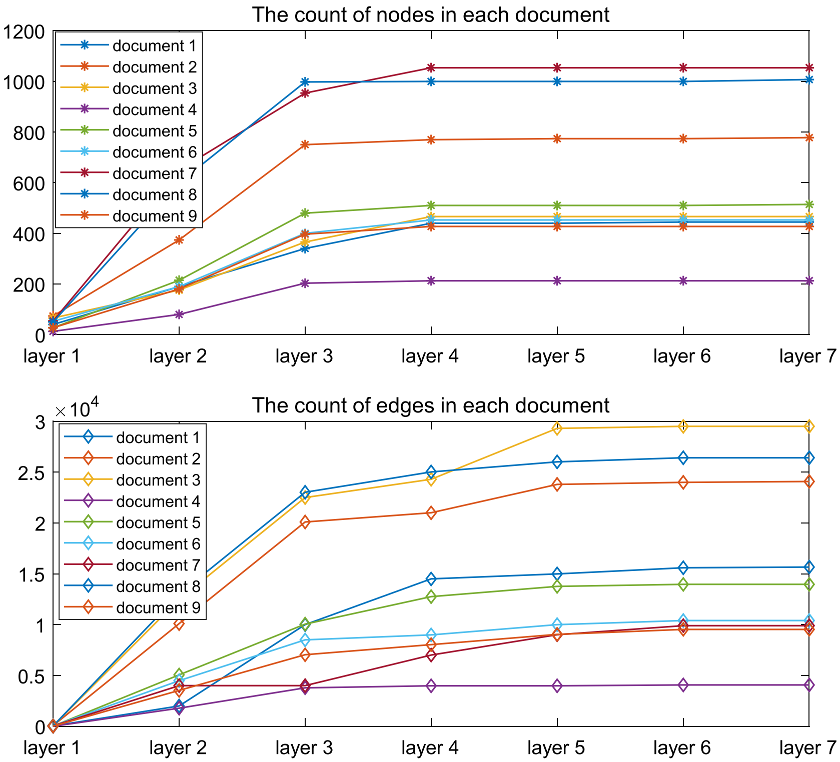

Fig. 5 Impact of different depth h values on the graphs

λ to 0.5, and both the content and the structure of feature terms should be considered when extracting a document's keywords.

Variations in depth

In addition, to compare different graph layers, we randomly selected one article from each category and calculated the number of nodes in the graphs with different depths h. The result indicates that the nodes in graphs constructed with 4 layers or more may capture the main informational content of the document. Therefore, the sampling depth h was set to 4 in this paper.

In Fig. 5, the node count in each document increases as the graph layer depth (or sampling depth h) grows, the node count stabilizes after reaching 4 layers. The edge count also increases as the layer depth grows and stabilizes after reaching layer 4. The weight values can be set differently to satisfy the varied topic extraction requirements of practical research.

For one article, the root node is selected automatically according to the maximum information gain in our method and shows relative stability. Therefore, the influencing factor of the structure of the graph is the sampling depth. If the sampling depth setting is only enough to extract part of the content of the document, then the keyword extraction results will lose information due to the lack of the extracted content; if the sampling depth is large enough, in fact, almost all the topic terms and relationships in the document will be extracted. Therefore, the hierarchical semantic graph for dephs 4 or more can completely express the stable correlation relationship among nodes.

Variations in keyword count

To verify the validity of the proposed method in dealing with the extraction of different numbers of keywords, another series of experiments are performed with keyword counts of 5, 7, 9, and 11. The extracted keywords are evaluated based on precision, recall, F-measure, MRR, and MAP. To prove the necessity of keyword collection extraction, we carried out 4 kinds (5, 7, 9, and 11) of keyword extraction for each category for performance evaluation. The final keyword extraction result in our method is shown in Table 3.

The experimental results in Table 3 show that the proposed method has good effectiveness and adaptability in handling the extraction of different numbers of keywords. For the changes in precision, recall, and F-measure with the number of keywords extracted, the precision shows an increasing trend as the number of keywords increases, achieving ideal results for all the keyword counts. The recall, in contrast, displays a decreasing trend, with better results when the keyword count is 5 or 7. In addition, MRR and MAP are most stable when the number of keywords is between 7 and 11, inclusive. It achieves its best effect when the number of keywords ranges from 5 to 9.

Generally, we observed that our model achieves better results on different group datasets across categories. Based on the comparative analysis of the results in each category, we note that documents in the “space,” “environment,” “agriculture”, and “computer” categories tend to have better performance. Returning to the dataset and analyzing the source documents, the documents in these categories tend to be long and have more paragraphs. This suggests that the proposed graph method performs well when applied to documents of high lengths and paragraphs.

Comparison with other methods

To further assess the performance of the proposed method, existing methods based on unsupervised learning were applied to the same dataset. The baselines selected for comparison in this experiment are as follows: (i) statistical models: TF-IDF, YAKE (Campos et al., 2018); (ii) graph-based models: SingleRank (Wan & Xiao, 2008), TopicRank (Bougouin et al. 2013), Topical PageRank (Sterckx et al., 2015), Multipartite Rank (Boudin., 2018); (iii) Embedding-based method: word2vec,Glove. The results are organized and reported as follows.

As seen in Fig. 6, the results for the baselines and the proposed model are detailed. Overall, we observed that the proposed model achieves the best results and significantly outperforms the baselines on most metrics. The F-measure value of our method exceeded 0.6, indicating better performance. The measures of the statistical models and graph-based models were less than 0.6. Compared with the statistical models, our proposed method makes up for the deficiency of semantic correlation caused by the assumption that words are independent of each other. Furthermore, compared with the SingleRank and TopicRank methods, the keyword extraction method in this paper realizes a further improvement. Our method has the advantage of considering the deep structural relationship between terms and has no dependence on clustering quality. Compared with word2vec, our method shows good performance in terms of precision and F-measure. For analysis and comparison, the

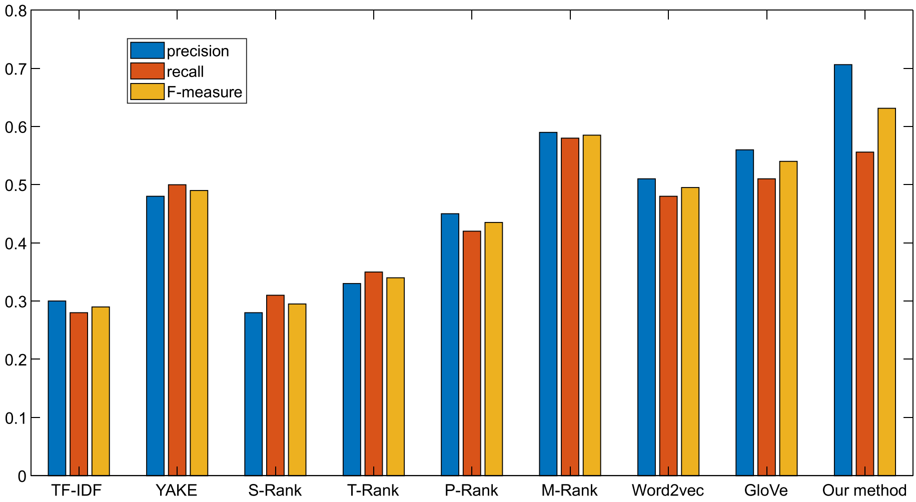

Fig. 6 Performance comparison in terms of different methods based on precision, recall and F-measure

ability of word2vec and Glove are general and does not use global cooccur information. Our method comprehensively integrates the content features with the intrinsic correlation structure and effectively represents keyword cohesion. This also means that the keyword collection selected by this method reveals the document content.

Statistical analysis

To analyze the relation of the keyword collection results to the content and structure of the documents, we perform statistical analyses on the distribution of keyword collection *K* and the specific feature terms (SFT) connected with these keywords based on all datasets. The statistics for analysis are set as follows: the ratio of node numbers (RNN), the ratio of node frequencies (RNF), the ratio of edge numbers (RNB), and the ratio of edge weights (REW).

RNN indicates the ratio of the connected node count to the total node count for graph G. The connected nodes (CN) here are the nodes in G that are connected to the terms in *K*. RNF is the ratio of the sum of CN frequencies to the sum of total node frequencies in G. RNB is the ratio of the connected edge count to the total edge count for G, where connected edges (CE) are edges connected to *K*. REW is the ratio of the sum of CE weights to the sum of edge weights in G.

Based on our datasets, the four aforementioned statistics are collected, and their respective maximum, minimum, and average values are listed, as shown in Fig. 7.

As the figure shows, RNN and RNF are the results of the statistical analysis based on the number and frequency of specific feature terms (SFT) connected with the extracted keyword collection; their mean values are 0.608 and 0.304, respectively. The results show that these SFTs are high both in the absolute count and frequency, which indicates that the extracted keyword combination plays an important role in the document and can effectively reflect the document's core content. RNB and REW are the results of the statistical analysis based on the number and weight of the SFT edges that connect with the extracted keyword collection; their mean values are 0.177 and 0.174, respectively. The results indicate that the number and weight of the SFT edges are both relatively high. Since the feature terms in

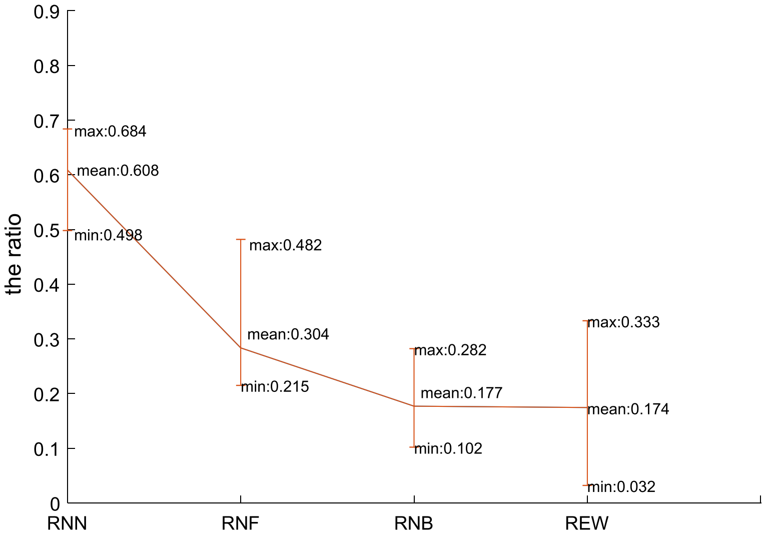

Fig. 7 Maximum, minimum and average values of RNN, RNF, RNB, and REW

these documents have a complex structure, the proportion shown here (nearly 20%) shows the importance of the extracted structure within the document and suggests that it reflects the core structural information of the document.

Next, we conduct a structural analysis for the specific feature terms based on different correlation intervals, with the latter again defined as $r = r(k_i, k_j)$. In this experiment, since the values of r range from 0 to 0.25, five intervals are delineated with thresholds of 0.2, 0.15, 0.1, 0.05, and 0. The number of specific feature terms (SFTs) connected with the extracted keyword combination in each interval was counted for each document as shown in Table 4.

Table 4 indicates the numerical distribution of specific feature terms connected to the extracted keyword combination in each document. For example, '102' in the table denotes the number of SFTs connected to the extracted keyword combination when r is [0.2,1) in paper 1. As seen in the table, the number of feature terms increases with the increase in range r, and the extent of the increase slows down gradually. By analyzing the results in each interval, we find that the number of SFTs increases the most between [0.2, 1) and [0.15, 1), with the smallest increase between [0.05, 1) and (0, 1). This shows that the distribution of feature terms in semantic graphs is concentrated around the keyword collection, which further illustrates the core role of the extracted keywords in contributing to the documents' content and structure.

In addition, a Kolmogorov–Smirnov test (K-S test) is used to quantify the test results and further evaluate the fixed-distribution assumption. The results of the K-S test are shown in Table 4.

Table 5 reports the average of the statistical values based on the K-S test, representing the logical value (H), P (p value), K-S test statistic (K-S stat), and critical value of the test (CV). The returned value of $H = 0$ and $p > 0.25$ indicates that the K-S test fails to reject the hypothesis at the default 5% significance level. This further supports the assumption that the data are approximately exponentially distributed. The exponential

Table 3 Keyword extraction performance for different keyword extraction

Group no	Evaluations	Precision	Recall	F-measure	MRR	MAP
Group 1 (Art)	5 keyword	0.707	0.464	0.55	0.414	0.673
	7 keyword	0.720	0.479	0.569	0.348	0.678
	9 keyword	0.711	0.48	0.554	0.301	0.696
	11 keyword	0.754	0.465	0.547	0.266	0.700
Group 2 (History)	5 keyword	0.729	0.433	0.533	0.413	0.677
	7 keyword	0.721	0.487	0.573	0.348	0.677
	9 keyword	0.686	0.505	0.563	0.301	0.678
	11 keyword	0.722	0.507	0.568	0.266	0.674
Group 3 (Space)	5 keyword	0.677	0.494	0.565	0.415	0.700
	7 keyword	0.768	0.423	0.535	0.346	0.768
	9 keyword	0.821	0.365	0.492	0.298	0.817
	11 keyword	0.868	0.325	0.451	0.264	0.859
Group 4 (Computer)	5 keyword	0.676	0.518	0.58	0.415	0.700
	7 keyword	0.778	0.449	0.56	0.346	0.768
	9 keyword	0.836	0.39	0.518	0.298	0.817
	11 keyword	0.886	0.346	0.482	0.264	0.859
Group 5 (Environment)	5 keyword	0.723	0.54	0.612	0.42	0.713
	7 keyword	0.79	0.445	0.56	0.354	0.766
	9 keyword	0.841	0.381	0.512	0.304	0.809
	11 keyword	0.884	0.337	0.473	0.265	0.845
Group 6 (Agriculture)	5 keyword	0.753	0.511	0.599	0.424	0.722
	7 keyword	0.792	0.478	0.586	0.370	0.756
	9 keyword	0.812	0.452	0.557	0.314	0.790
	11 keyword	0.853	0.426	0.535	0.275	0.811
Group 7 (Economy)	5 keyword	0.74	0.514	0.596	0.413	0.707
	7 keyword	0.782	0.503	0.603	0.348	0.747
	9 keyword	0.800	0.485	0.573	0.301	0.779
	11 keyword	0.840	0.46	0.563	0.267	0.798
Group 8 (Politics)	5 keyword	0.761	0.46	0.610	0.414	0.724
	7 keyword	0.751	0.493	0.622	0.348	0.711
	9 keyword	0.736	0.515	0.625	0.301	0.711
	11 keyword	0.771	0.513	0.642	0.266	0.724
Group 9 (Sports)	5 keyword	0.758	0.509	0.633	0.428	0.724
	7 keyword	0.773	0.499	0.636	0.355	0.735
	9 keyword	0.789	0.491	0.64	0.305	0.762
	11 keyword	0.830	0.469	0.649	0.268	0.786

distribution of the result indicates that the smaller the term correlation r is, the greater the number of terms extracted; conversely, the fewer the number of terms extracted. The distribution result reveals a universal law of keyword extraction, namely, the real highly relevant terms are often the core words with a small percentage in a document.

Table 4 Partial results display of SFT numerical distribution

Paper no	[0.2,1)	[0.15,1)	[0.1,1)	[0.05,1)	(0,1)
Paper 1	**102**	411	562	644	689
Paper 2	108	577	849	1,112	1,229
Paper 3	131	372	534	572	611
Paper 4	43	122	144	146	169
Paper 5	163	558	693	756	799
…	…	…	…	…	…

Table 5 K-S test on numerical data distribution

K-S test	H	*p* Value	K-S stat	CV
Mean value	0	0.2998	0.3005	0.4112

Discussion

The experimental results in this paper show that the method proposed herein can extract a keyword collection that captures a document's core content by analyzing the hierarchical structure of its feature terms. Compared with existing keyword extraction methods, this method displays higher precision, recall and F-measure, metrics that speak to its superior effectiveness.

Two main innovations are presented in this paper: (1) hierarchical extraction of feature terms and semantic graph construction and (2) keyword collection extraction based on the hierarchical relevance of feature terms. A semantic graph such as the one described herein can effectively reveal the hierarchical structure distribution of feature terms. The semantic graph and topic feature term extraction model constructed in this paper can not only distinguish the topic feature difference relationship between different paragraphs but also reveal the structural topic feature association relationship between different paragraphs. Furthermore, the extracted keywords can reveal cohesion between keywords by mining the contribution degree of the keyword collection.

Traditional keyword extraction methods, such as TD-IDF (Aizawa, 2003), measure the word importance by only frequency, but such a metric is not complete enough to represent a word in context. The correlation between terms in this paper is obtained through global term-term paragraph distribution statistics, getting rid of single word frequency statistics. To a certain extent, it overcomes the shortcomings of traditional keyword extraction algorithms based only on word frequency. Since the correlation is calculated based on paragraph distribution, the terms with lower frequency may obtain a higher word correlation when they have similar paragraph distribution in the specific data distribution. Specifically, methods based on statistical frequency cannot effectively reflect the semantic and correlational characteristics of words. The keyword extraction method proposed in this paper deeply analyses the content and structure of feature terms and thereby effectively reveals the hierarchical correlation between feature terms. Other semantic-based keyword extraction methods, such as LDA (Blei et al., 2003; Liu et al., 2010), use documents' implicit semantic information to extract keywords, but the keywords extracted from the topic model are relatively broad and cannot reflect the document themes. By mining the intrinsic hierarchical correlation between feature terms and analyzing the extracted keywords as a set, our method yields results that better cover the document themes.

The existing semantic graph construction method considers that if two feature terms are correlated, there is an edge between their two nodes (Garg & kumar, 2018; Pujara et al., 2013). The graph construction described here, in contrast, relies on the hierarchical extraction of nodes based on the correlation between feature terms. This means that if the correlation between two words is below a certain threshold, no relationship is deemed to exist between the two nodes. In addition, this paper simultaneously considers the similarities and differences between feature terms, unlike other studies, which utilize semantics and grammatical similarity between feature terms (Blanco & Lioma, 2012; Rose et al., 2010; Tixier et al., 2016). Moreover, keyword selection in existing research is based on ranking node importance as derived from connectivity and centrality (Rafiei-Asl & Nickabadi, 2017; Ravinuthala & Ch, 2016; Ying et al., 2017). However, the keywords that best express the document topic must not only include highly important individual nodes but also form cohesive structures among these nodes. We refactor node importance and selecting the keyword collection with the maximum contribution as the keyword result. This reveals the cohesion of the results and improved the accuracy of keyword extraction results.

Conclusion

In this paper, we proposed a keyword extraction method based on a hierarchical semantic graph. First, the initial semantic graph was constructed based on the correlation between feature terms; Next, a candidate keyword graph was generated based on the import intensity of the semantic graph's nodes. Finally, the keyword collection was evaluated as a set by calculating the joint probability of candidate keywords. The algorithm proposed here considers the contextual environment of terms as well as the internal hierarchical structure between them; it thus stands to overcome the shortcomings of the traditional keyword extraction methods based on literal matching. In addition, through mining deep implicit structures between terms, the proposed method reveals the hierarchical correlation mechanism between keywords and improves the accuracy of extracted keywords. Experimental results showed that the proposed method outperforms statistical models, graph-based models and embedding-based method, in terms of precision, recall, and F-measure and ranking quality measures. Statistical analysis also revealed that the proposed algorithm has a certain reference value for the research and application of keyword extraction.

The present study does, however, have some limitations. In the graph construction process, the correlation weight, and regulating variable may influence the experimental results. Furthermore, the correlations between terms in our approach rely on the term distribution in each paragraph. This means that our method works for documents with several paragraphs and may be limited if one article has only one or two paragraphs. In the future, we will consider using sentences instead of paragraphs as the basic segmentation unit in response to the challenge of different documents. Finally, although this paper considers the highest aggregate contribution of feature terms, it does not reveal the subdivision meaning of individual keywords, such as those describing a paper's field, perspective, and method.

Acknowledgements The authors are grateful to the reviewers for their valuable suggestions on how to improve the paper. We thank Dr. Mouda for help with the manuscript writing and Dr. Haobo for the proofreading. The author acknowledges the support by the Project No. 72074117, 71972090, 72274040 funded by National Natural Science Foundation of China; Project No. 20KJB630012, 2020SJA0344 funded by University Science Research Project and Philosophy and Social Science Research of Jiangsu Province. Project

No. 2021SJZDA153 funded by the Significant Project of Jiangsu College Philosophy and Social Sciences Research. We also thank Dr. Minwei for advice on experimental design.

References

Abilhoa, W. D., & De Castro, L. N. (2014). A keyword extraction method from twitter messages represented as graphs. *Applied Mathematics and Computation, 240*, 308–325.

Alqaryouti, O., Khwileh, H., Farouk, T., Nabhan, A., & Shaalan, K. (2018). Graph-based keyword extraction. In *Intelligent Natural Language Processing: Trends and Applications* (pp.159–172). Springer.

Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. *Information Processing & Management, 39*(1), 45–65.

Beliga, S., Kitanović, O., Stanković, R., & Martinčić-Ipšić, S. Keyword Extraction from Parallel Abstracts of Scientific Publications.(*2017*). In *International KEYSTONE Conference on Semantic Keyword-Based Search on Structured Data Sources,* pp. 44–55.

Beliga, S., Meštrović, A., & Martinčić-Ipšić, S. (2015). An overview of graph-based keyword extraction methods and approaches. *Journal of Information and Organizational Sciences, 39*(1), 1–20.

Biswas, S. K., Bordoloi, M., & Shreya, J. (2018). A graph based keyword extraction model using collective node weight. *Expert Systems with Applications, 97*, 51–59.

Blanco, R., & Lioma, C. (2012). Graph-based term weighting for information retrieval. *Information Retrieval, 15*(1), 54–92.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research., 3*, 993–1022.

Boudin, F. (2018). Unsupervised keyphrase extraction with multipartite graphs. *arXiv Preprint arXiv:1803.08721*. https://doi.org/10.48550/arXiv.1803.08721

Bougouin A, Boudin F, Daille B. Topicrank: Graph-based topic ranking for keyphrase extraction. In *International joint conference on natural language processing* pp. 543–551.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems, 30*(1–7), 107–117.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A. M., Nunes, C., & Jatowt, A. (2018). A Text Feature Based Automatic Keyword Extraction Method for Single Documents. In *European Conference on Information Retrieval*, 684–691.

Chidambaram, S., & Srinivasagan, K. (2016). Optimization approach for feature selection and classification with support vector machine. *Computational Intelligence in Data Mining, 1*, 103–111.

Duari, S., & Bhatnagar, V. (2019). sCAKE: Semantic connectivity aware keyword extraction. *Information Sciences, 477*, 100–117.

El-Beltagy, S. R., & Rafea, A. (2009). KP-Miner: A keyphrase extraction system for English and Arabic documents. *Information Systems, 34*(1), 132–144.

Figueroa, G., Chen, P.-C., & Chen, Y.-S. (2018). RankUp: Enhancing graph-based keyphrase extraction methods with error-feedback propagation. *Computer Speech & Language, 47*, 112–131.

Garg, M., & Kumar, M. (2018). The structure of word co-occurrence network for microblogs. *Physica a: Statistical Mechanics and Its Applications, 512*, 698–720.

Gopan E , Rajesh S , Gr V , et al. (2020). Comparative Study on Different Approaches in Keyword Extraction. *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*. 2020: pp. 70–74.

Hashemzahde, B., & Abdolrazzagh-Nezhad, M. (2020). Improving keyword extraction in multilingual texts. *International Journal of Electrical and Computer Engineering, 10*(6), 5909.

Hulth A, Megyesi B. (2006). A study on automatically extracted keywords in text categorization. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. pp. 537–544.

Jose, L. M., & Rahamathulla, K. (*2016*). A semantic graph based approach on interest extraction from user generated texts in social media. In *Data Mining and Advanced Computing (SAPIENCE), International Conference on,* 101–104.

Kumar, M., & Rehan, P. (2021). Graph node rank based important keyword detection from Twitter. *Applied Computing and Informatics, 17*(2), 194–209.

Litvak, M., Last, M., Aizenman, H., Gobits, I., & Kandel, A. (2011). DegExt—A language-independent graph-based keyphrase extractor. In *Advances in Intelligent Web Mastering–3*. 121–130.

Liu, Z., Li, P., Zheng, Y., & Sun, M. (2009). Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1* (pp. 257–266).

Liu, Z., Huang, W., Zheng, Y.,et al. (2010). Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 conference on empirical methods in natural language processing,*(pp. 366–376).

Mahata, D., Kuriakose, J., Shah, R., & Zimmermann, R. (2018, June). Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 2 (Short Papers). 634–639.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 401–411).

Mothe, J., Ramiandrisoa, F., & Rasolomanana, M. (2018). Automatic keyphrase extraction using graph-based methods. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing* (pp. 728–730). https://doi.org/10.1145/3167132.3167392

Naidu, R., Bharti, S. K., Babu, K. S., & Mohapatra, R. K. (2018). Text summarization with automatic keyword extraction in Telugu e-newspapers. *Smart Computing and Informatics, 1*, 555–564.

Nasar, Z., Jaffry, S. W., & Malik, M. K. (2018). Information extraction from scientific articles: A survey. *Scientometrics, 117*(3), 1931–1990.

Nguyen, Thuy Dung, & Min-Yen Kan.(2007) "Keyphrase extraction in scientific publications." *International conference on Asian digital libraries*. Springer, Berlin, Heidelberg: pp. 317–326.

Onan, A., Korukoğlu, S., & Bulut, H. (2016). Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications, 57*, 232–247.

Papagiannopoulou, E., & Tsoumakas, G. (2019). A review of keyphrase extraction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. https://doi.org/10.1002/widm.1339

Pu, X., Jin, R., Wu, G., Han, D., & Xue, G.-R. (2015).Topic modeling in semantic space with keywords. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 1141–1150.

Pujara, J., Miao, H., Getoor, L., & Cohen, W. (*2013*). Knowledge graph identification. In *International Semantic Web Conference,* pp. 542–557.

Qian, Y., Santus, E., Jin, Z., Guo, J., & Barzilay, R. (2018). GraphIE: A graph-based framework for information extraction. *arXiv Preprint arXiv:1810.13083*. https://doi.org/10.48550/arXiv.1810.13083

Rafiei-Asl, J., & Nickabadi, A. (2017). TSAKE: A topical and structural automatic keyphrase extractor. *Applied Soft Computing, 58*, 620–630.

Ravinuthala, M. K. V., & Ch, S. R. (2016). Thematic text graph: A text representation technique for keyword weighting in extractive summarization system. *International Journal of Information Engineering and Electronic Business, 8*(4), 18.

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory*. https://doi.org/10.1002/9780470689646.ch1

Siddiqi, S., & Sharan, A. (2015). Keyword and keyphrase extraction techniques: A literature review. *International Journal of Computer Applications*. https://doi.org/10.5120/19161-0607

Sterckx, L., Demeester, T., & Deleu, J. (2015). Topical word importance for fast keyphrase extraction. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 121–122).

Tixier, A., Malliaros, F., & Vazirgiannis, M. (2016). A graph degeneracy-based approach to keyword extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* pp. 1860–1870.

Treeratpituk, P., Teregowda, P., Huang, J., & Giles, C. L. (2010). Seerlab: A system for extracting key phrases from scholarly documents. In *Proceedings of the 5th international workshop on semantic evaluation,* pp. 182–185.

Tutkan, M., Ganiz, M. C., & Akyokuş, S. (2016). Helmholtz principle based supervised and unsupervised feature selection methods for text mining. *Information Processing & Management, 52*(5), 885–910.

Vanyushkin, A., & Graschenko, L. (2020). Analysis of text collections for the purposes of keyword extraction task. *Journal of Information and Organizational Sciences, 44*(1), 171–184.

Wan, X., & Xiao, J. (2008). CollabRank: towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics* (pp. 969–976).

Wang, R., Liu, W., & McDonald, C. (2015, June). Using word embeddings to enhance keyword identification for scientific publications. *In Australasian Database Conference*. 257–268.

Wang, H., Ye, J., Yu, Z., et al. (2020). Unsupervised keyword extraction methods based on a word graph network. *International Journal of Ambient Computing and Intelligence, 11*(2), 68–79.

Witten, I. H., et al. (2005). *Kea: Practical automated keyphrase extraction. Design and usability of digital libraries: Case studies in the asia pacific* (pp. 129–152).

Xie, F., Wu, X., & Zhu, X. (2017). Efficient sequential pattern mining with wildcards for keyphrase extraction. *Knowledge-Based Systems, 115*, 27–39.

Xu, Z., & Zhang, J. (2021). Extracting keywords from texts based on word frequency and association features. *Procedia Computer Science, 187*, 77–82.

Yang, L., Li, K., & Huang, H. (2018). A new network model for extracting text keywords. *Scientometrics, 116*, 339–361.

Ying, Y., Qingping, T., Qinzheng, X., Ping, Z., & Panpan, L. (2017). A graph-based approach of automatic keyphrase extraction. *Procedia Computer Science, 107*, 248–255.

Zhang, K., Xu, H., Tang, J., & Li, J. (2006). Keyword extraction using support vector machine. In*Advances in Web-Age Information Management: 7th International Conference, WAIM 2006, Hong Kong, China, June 17-19, 2006. Proceedings 7* (pp. 85–96). Springer Berlin Heidelberg.

Zhang, C. (2008). Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems, 4*(3), 1169–1180.