Complex & Intelligent Systems (2022) 8:1–12
https://doi.org/10.1007/s40747-021-00343-8

ORIGINAL ARTICLE

A patent keywords extraction method using TextRank model with prior public knowledge

Zhaoxin Huang[1,2] · Zhenping Xie[1,2]

Received: 30 November 2020 / Accepted: 17 March 2021 / Published online: 29 March 2021

Abstract

For large amount of patent texts, how to extract their keywords in an unsupervised way is a very important problem. In existing methods, only the own information of patent texts is analyzed. In this study, an improved TextRank model is proposed, in which prior public knowledge is effectively utilized. Specifically, two following points are first considered: (1) a TextRank network is constructed for each patent text, (2) a prior knowledge network is constructed based on public dictionary data, in which network edges represent the prior interpretation relationship among all dictionary words in dictionary entries. Then, an improved node rank value evaluation formula is designed for TextRank networks of patent texts, in which prior interpretation information in prior knowledge network are introduced. Finally, patent keywords can be extracted by finding top-k node words with higher node rank values. In our experiments, patent text clustering task is used to examine the performance of proposed method, wherein several comparison experiments are executed. Corresponding results demonstrate that, new method can markedly obtain better performance than existing methods for patent keywords extraction task in an unsupervised way.

Keywords Extraction · Patent text · Prior knowledge network · TextRank model

Introduction

For more and more patent texts, how to mine their contents to effectively obtain valuable patent information has aroused widespread concern [1, 2]. Generally, patent contents can be well represented by some key term words, also called patent keywords. Then, these keywords can be widely used in text mining such as automatic summary generation [3], patent novelty discovery [4], text clustering and classification [5, 6].

Patent texts, as a type of semi-structured texts, usually have relatively regular structures about their paragraphs and sentences. Thus, TextRank [7] as a graph-based analysis model has been effectively used to extract patent keywords [8]. For existing TextRank methods, they use the PageRank [9] formula to calculate node rank values based on the co-occurrence relationship among all possible words. However, these methods did not consider the original differences of term node importance over public common knowledge. Therefore, an improved TextRank model is proposed in this study by introducing a prior knowledge network, which is called as PrTextRank in this text.

In PrTextRank, a patent text is first modeled as a classical TextRank network, and a public dictionary data are modeled as a prior knowledge network. In the prior knowledge network, node weights are also computed by PageRank formula, and the edge weights are computed by means of the co-occurrence degrees among node words in dictionary entries. Then, the prior information in prior knowledge network is integrated into patent TextRank networks, and a new evaluation method of node rank value is designed. Finally, patent keywords can be extracted by finding the top-k node words with higher node rank values like in standard TextRank model.

The main contributions of this study can be summarized as follows:

✉ Zhenping Xie
xiezp@jiangnan.edu.cn

[1] School of Artificial Intelligence and Computer Science, Jiangnan University, No. 1800 Lihu Avenue, Wuxi 214122, Jiangsu, People's Republic of China

[2] Jiangsu Key Laboratory of Media Design and Software Technology (Jiangnan University), No. 1800 Lihu Avenue, Wuxi 214122, Jiangsu, People's Republic of China

(i) An improved TextRank model with prior public knowledge is effectively proposed for patent text keywords extraction.
(ii) Several experiments on patent text clustering tasks using several clustering methods are performed, in which, several compared methods are considered including our proposed method, Term Frequency-Inverse Document Frequency (TF-IDF), TextRank and TopicalPageRank. According to experimental results, good performance of our proposed method can be indicated.
(iii) An extended experimental analysis is also executed on other types of text including news text and food popular texts. Corresponding experimental results also display the availability of our proposed method.

The rest of this text is structured as follows. The second section briefly reviews the related works. The third section describes the proposed TextRank model. The fourth section gives our experimental analysis and results. In the fifth section, we provide a summary of our work.

Related work

Keywords extraction

Keywords extraction is the foundation of many text mining tasks and has been widely studied by many researchers. Wherein, supervised and unsupervised strategies are two basic categories. The supervised keywords extraction methods can be regarded as a binary classification processing, which needs labeled corpus data to train classification functions to obtain good performance, for example, the methods based on decision tree [10], Support Vector Machines (SVM) [11], neural networks [12] and so on. In recent years, with the deepening of deep learning research, many keywords extraction techniques based on neural network emerged. Zhang et al. [13] proposed a target center-based Long Short-Term memory (LSTM) model (TC-LSTM) to achieve performance improvement. She et al. [14] proposed a deep neural semantic network (DNSN). Feng et al. [15] used reinforcement learning and deep learning to extract entities and relationships from texts, and used bidirectional LSTM to realize preliminary entity extraction. Then, Tree-LSTM was designed to capture the most important information mentioned in the relationship.

However, the supervised keywords extraction methods rely too much on labeled corpus. Because manual labeling is time-consuming and laborious, supervised methods will be limited in many application scenes.

At present, more and more researchers are focusing on unsupervised keywords extraction problems. Such type of methods usually designs different scoring criteria to rank candidate keywords, and to extract the top-k words as keywords. Generally speaking, unsupervised extraction methods may be divided into three categories: statistical methods, word graph methods and Latent Dirichlet Allocation (LDA) methods.

The most classical statistical method is TF-IDF [16], which has strong applicability, but it mainly relies on word frequency information, which leads to ignoring the semantic features in the text.

Another kind of famous unsupervised methods are word graph methods. Inspired by the great success of PageRank algorithm and its wide application, Mihalcea et al. [7] proposed the TextRank method in 2004, which constructs an undirected weighted graph and uses the PageRank iterative calculation formula to calculate the importance of nodes. However, the graph-based keyword extraction algorithm needs many iterations, which increases the complexity of calculation. Gollapalli et al. [17] proposed CiteTextRank method, which uses citation networks to enhance the information content of text graph. Florescu et al. [18] put forward PositionRank method by adding word position information to the PageRank model. In addition, Devika et al. [19] proposed Semantic graph-based Keywords Extraction Method (SKEM), which can effectively extract keywords by means of semantic information and graph indices.

By calculating the similarity between candidate keywords and topics, the extraction method based on LDA could be proposed. As early as 2010, Liu of Tsinghua University put forward the algorithm of TopicalPageRank (TPR) [20], in which the PageRank score of each candidate keyword under corresponding topic is evaluated. However, above method based on LDA is largely affected by the distribution of training topics. In addition, the number of topics should be adjusted manually in advance.

Patent analysis based on text mining

With the standardization and specialization of patent texts, text mining has been widely used in patent analysis. When applying text mining methods in the field of patent analysis, most people have paid attention to meaningful keywords extraction. For example, Yang et al. [21] used regular expression pattern matching techniques to extract semantic information from patent claims. Noh et al. [22] took four different factors to determine the best keyword selection and processing strategy for patent texts.

Technology evolution analysis, new technology discovery, and patent search are all important patent analysis tasks. Madani et al. [23] used CiteSpace for bibliometric analysis and cluster analysis of keyword networks to analyze patent evolution problems. Park et al. [24] perform the technology opportunity discovery by means

of comprehensive analysis of patent classification and collaborative filtering. Yanagihori et al. [25] created an extended dictionary of word meaning, and applied compound noun analysis to realize similar patent search.

Clustering algorithm

At present, the commonly used clustering methods are partition methods, hierarchical methods, density-based methods, graph clustering methods, etc.

Partition-based clustering algorithms iteratively divide the data into different clusters until the distances between points in the same cluster are close enough while the distances between points in different clusters are far enough. Among them, the most classical clustering algorithm based on partition is k-means [26, 27]. K-means algorithm is simple and efficient, but it needs to preset the number of clusters and it is sensitive to the selection of initial cluster centers.

Hierarchical clustering algorithms combine the nearest points into one cluster, and then combines the nearest clusters into one big cluster, until all points form one cluster. Although these methods do not need to set the number of clusters in advance, its computational complexity is very high.

Density-based clustering algorithms can realize irregular shape clustering by density estimation, and decide whether to continue clustering according to whether the point densities in a region exceed a threshold. A classical algorithm is Density-Based Spatial Clustering of Application with Noise (DBSCAN) [28]. Although this algorithm can realize the clustering of irregular shapes, its performance depends too much on conditional parameters.

Based on graph theory, graph clustering algorithms regard data as points with connected edges in a graph space, and realize clustering by cutting graphs. A classic algorithm is spectral clustering [29]. Spectral clustering algorithm depends on the input of similarity matrix, which is easy to implement, but it is also sensitive to initial parameter selection.

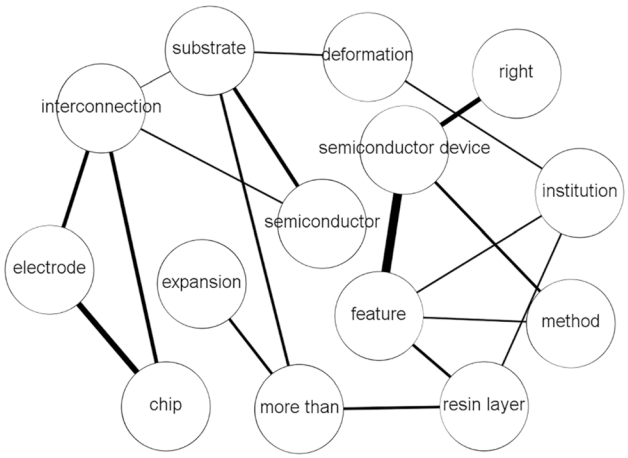

Fig. 2 An example of patent TextRank network

The proposed method

Overview

Considering the introduction of prior knowledge network, a new model framework is designed in this study, as shown in Fig. 1.

In this study, first, the patent text is preprocessed to obtain candidate keywords. Wherein, the accuracy of identified candidate keywords will directly affect the quality of finally extracted keywords [30]. At the same time, the public dictionary data are also preprocessed. Then, each patent text is modeled as a TextRank network, and a prior knowledge network is constructed based on public dictionary data. Further, an improved node rank value evaluation formula is designed by combining the prior information in prior knowledge network. Finally, the top-k nodes with higher node values are extracted as patent keywords.

TextRank model

The TextRank model is a typical graph-based keywords extraction method inspired by the PageRank. In this model, the text is modeled as an undirected weighted graph $G = (V, E, W)$, in which the candidate keywords are

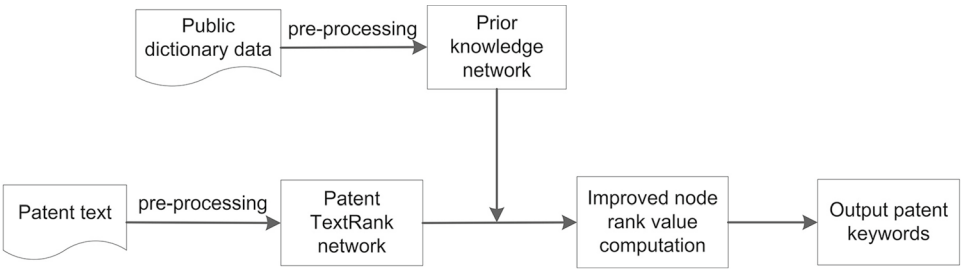

Fig. 1 Overview of PrTextRank method

regarded as the node set V, and the co-occurrence relationship between two words in a sliding window is regarded as an edge in E. W represents the time of co-occurrence with respect to E. Figure 2 shows an example of one patent TextRank network. The size of nodes in the graph is same, which means that all nodes have same initial weights. The thickness of an edge represents the value of W, and thicker edges denote larger W values. It can be seen from Fig. 2 that the edge between "semiconductor device" and "feature" is the thickest, which indicates that above two words appear most frequently in same sliding windows.

Inspired by the PageRank principle, the iterative calculation formula (1) is introduced to compute node weights.

$$S(V_i) = (1-d) + d \sum_{V_j \in In(V_i)} \frac{W_{ji}}{\sum_{V_k \in Out(V_j)} W_{jk}} S(V_j) \tag{1}$$

wherein, d is a damping coefficient of iterative computation, and may take 0.85 as default.$\mathrm{In}(V_i)$ represents the set of nodes pointing to V_i, $\mathrm{Out}(V_j)$ is the set of nodes pointed by V_j. The formula shows that the weight of a node V_i depends on the edge weight from V_j to V_i and the sum of edge weights from node V_j to other nodes.

TextRank model only uses the information of text itself, which is suitable for general fields, but ignores prior text characteristics for some specific applications.

Prior knowledge network

Usually, the entries in a public dictionary have been carefully constructed by domain experts. They should cover a wide range of fields and be authoritative. Based on above considerations, a directed prior knowledge network can be constructed based on public dictionary data, in which network nodes represent dictionary words, and network edges represent the explanatory relations among dictionary words.

A partial prior knowledge network is shown in Fig. 3. The node sizes in the network are set according to their in-degree values. We believe that the more times a node is interpreted by, the more possible it is a professional entry, and the more specific meaning it represents. It can be seen from the figure that the word in-degree value of "electronic computer" is very high, reflecting that this word has been explained more times and its content is more specific.

When constructing network edges, we record the co-occurrence times of edges between two dictionary words. After computing the edge weights (above co-occurrence time values) in prior knowledge network, node weights ($nw_{PKN}(i)$ for node i) could be further calculated using the PageRank iterative equation.

Prior keywords importance

In traditional TextRank model, all candidate keywords are assigned a same initial importance value. In this study, we think that some prior information of keywords can be considered for a patent text under a public dictionary.

First of all, we introduce the classical TF-IDF calculation method. Term frequency (TF) refers to the frequency with which a certain word appears in a given text document. The formula is as follows:

$$\mathrm{TF}_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \tag{2}$$

where n_{ij} represents the number of occurrences of the word t_i in document d_j, and the denominator represents the total number of occurrences of all words in document d_j.

In [31], the Inverse Document Frequency (IDF) is introduced to describe the frequency of documents containing a certain word in the corpus. If a few documents contain a certain keyword i, it shows that the keyword i has a good discrimination ability, and its IDF value is higher. The concrete formula for IDF is,

$$idf_i = \log \frac{|D|}{|\{j : t_i \in d_j\}| + 1} \tag{3}$$

where $|D|$ represents the number of documents in the corpus. The denominator in formula (3) represents the number of documents containing the word t_i, and adding 1 is to prevent meaningless 0 value.

Here, we believe that the information in prior knowledge network may be used to define different initial importance values of candidate keywords. According to general cognition, the words that appear in the definitions of other words may be more popular words, and their possibility as keywords will be lower. In contrast, other words will have higher possibility values of being keywords.

Inspired by above idea, we may introduce a node popularity for each node word in TextRank network. Concretely, it can be defined as follows:

$$pd_i = \sum_{n \in as(i)} nw_{PKN}(i,n) \tag{4}$$

wherein, $nw_{PKN}(i,n)$ represents the weights of n nodes associated with patent node word v_i in prior knowledge network. By combining above idea, it can be known that, the larger pd_i is, the more times of patent node word v_i explains other words, so the lower the importance of patent node word v_i is. Then, the prior importance of patent node word v_i in document D_j can be introduced as follows:

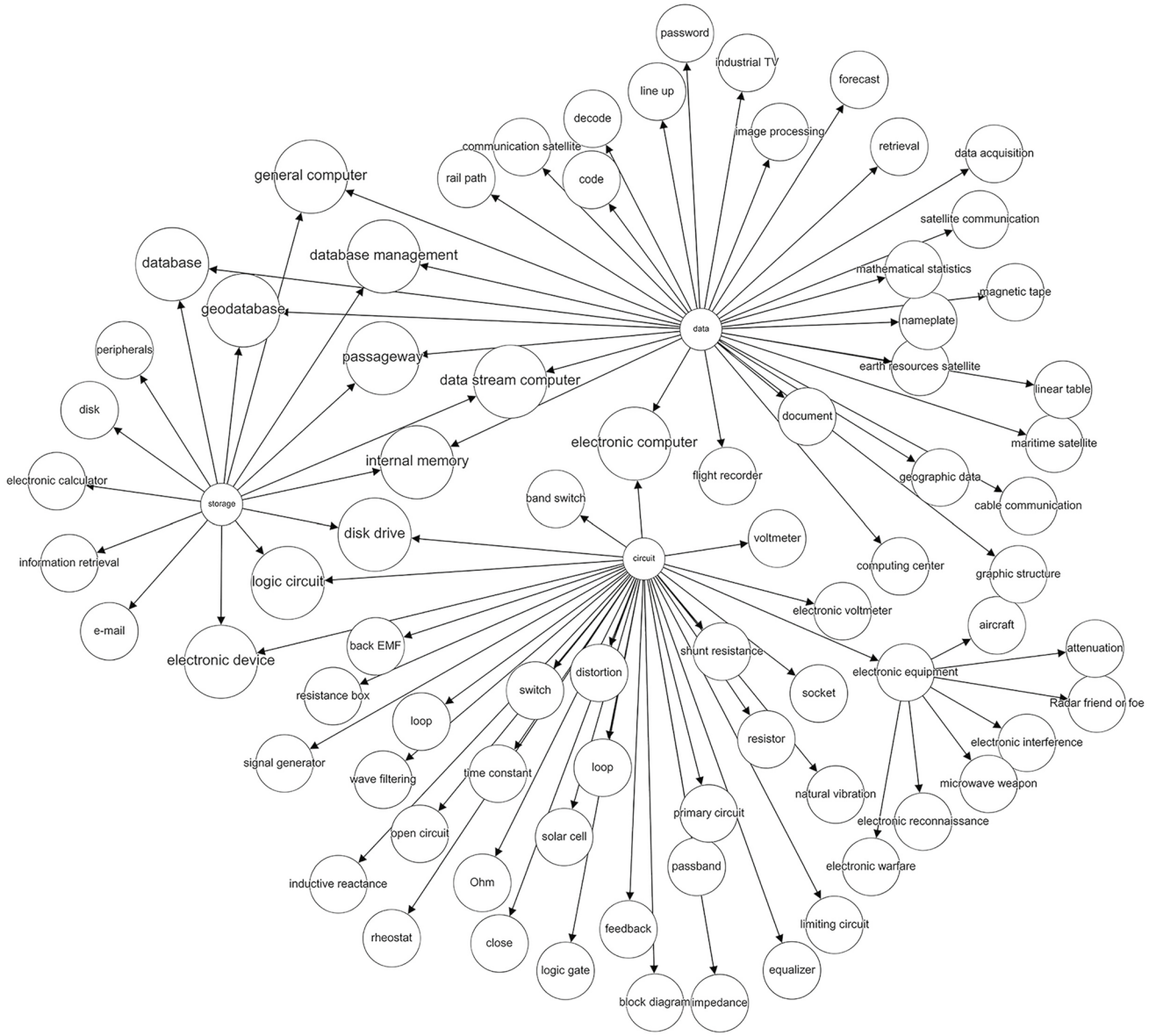

Fig. 3 A part of prior knowledge network

$$pi(v_i, D_j) = \frac{tf_{ij} \cdot idf_i}{\sqrt{pd_i + 1}} \tag{5}$$

So, we may think that, if any candidate word may be a valid keyword, then its prior importance should be higher.

Transfer factor computation

In TextRank model, to compute node rank values, edge weights also called transfer factor values on edges should be computed according to practical problem requirements.

On one hand, we consider that the transfer factor value on an edge is related to the co-occurrence frequency $f_{n'}$ of two node words n and n' connected by that edge.

On the other hand, we may use some prior information in prior knowledge network to calculate the value of transfer factor. Here, associative memory strategy is employed. Associative memory [32] is the association of new information and known things, and it is the core function of human brain. It can be believed that, the learning processing of human brain [33] may be the process of neurons' generation, deletion and association, also the process of associative memory. So, if each entity in the prior knowledge network is regarded as a

neuron, and a connection between two entities is regarded as an association relationship, we may calculate the associative relationship between two entities by calculating their connection strength. A concrete formula is proposed in [34], and its form is:

$$U_{n\prime}^{p}=\begin{cases}\sum\limits_{1\le i\le M}\sum\limits_{1\le j\le N}\sum\limits_{1\le k\le Co}\frac{1}{I_n\left(\left\langle n^{ijk},n\prime^{ijk}\right\rangle\right)-I_{n\prime}\left(\left\langle n^{ijk},n'^{ijk}\right\rangle\right)} & n\to n'\\ \frac{1}{10} & n'\end{cases} \tag{6}$$

where $\left\langle n^{ijk},n\prime^{ijk}\right\rangle$ denotes a connection between two nodes n and $n\prime$ in prior knowledge network, and Co represents the co-occurrence times of two node words in dictionary entries. I_n and $I_{n\prime}$ respectively indicate the relative position index value in that sentence. Besides, M represents maximum associative jump number of connecting two nodes n and $n\prime$, and N is the total number of all nodes in prior knowledge network. Here, $M=2$ is set as default for low computing complexity.

In addition, in formula (6), if there is no valid associative access within M steps, it is considered that there is no valid association between two node words, then $U_{n\prime}^{p}$ is set to be 1/10. It means that, if they have prior associative memory information for two node words, and two node words appear in a same sliding window in given patent text, then their transfer factor value should be higher.

Finally, we may introduce following equation of transfer factor calculation into our improved TextRank model.

$$W_{n\prime}^{p}=f_{n\prime}\cdot U_{n\prime}^{p} \tag{7}$$

Novel keywords rank value computation and extraction

According to above description, we may directly introduce following improved node rank value computing formula.

$$S_p\left(v_i,D_l\right)=(1-d)pi(v_i,D_l)+d\sum_{v_j\in In(v_i)}\frac{W_{ij}^{p}}{\sum_{v_k\in Out(v_j)}W_{jk}^{p}}S_p(v_j,D_l) \tag{8}$$

wherein, d is a damping coefficient same as in original TextRank model. $pi(v_i,D_l)$ denotes prior importance computed in formula (5). W_{ij}^{p} represents the transfer factor value defined in formula (7).

According to above discussion, the method process of PrTextRank can be concluded as follows.

Algorithm Keywords Extraction PrTextRank Method

INPUT: Pre-processed candidate keywords list CandWordsList, Number of keywords k

OUTPUT: KeywordsList

1: Initialize keywords set KeywordsList = [], and construct a prior knowledge network according given dictionary data
2: For each patent text, construct a TextRank network
3: Perform following steps to obtain node rank values until rank values to be converged
4: **for** i = 0 to length (CandWordsList) **do**
5: Calculate prior importance using formula (5)
6: **for** j = 0 to length (CandWordsList) **do**
7: obtain transfer factor between node CandWordsList [i] and CandWordsList [j] using equation (6);
8: **end for**
9: Update S_p(CandWordsList[i])
10: **end for**
12: add top-k keywords in KeywordsList
11: **return** KeywordsList

Experiments and analysis

First, experimental datasets and performance evaluation indices are introduced. Then, the performance results of four keyword extraction methods under three clustering methods will be reported. Finally, the availability of PrTextRank will be further tested on another dataset.

Datasets

The experimental patent text corpus is provided by Changzhou Baiteng Technology Company. According to the IPC classification standard, we use three categories of patent texts under the electrical category, each of which has 1,000 patent texts, a total of 3000 patent texts are used as dataset I.

As the typical representative of semi-structured texts, patent texts not only contain structured information such as application date and IPC classification number, but also contain unstructured information such as title indicating technique focus, abstract summarizing technique contents and claims revealing detailed technical scope. In this study, the title, abstract and claims of patent texts are used.

In addition, this study uses the data set used in paper [35], in which 953 Sohu sports news texts are included. At the same time, 953 texts were captured from Foodbk [36] as another text set. Such, 1906 texts of above two types of texts are taken as dataset II.

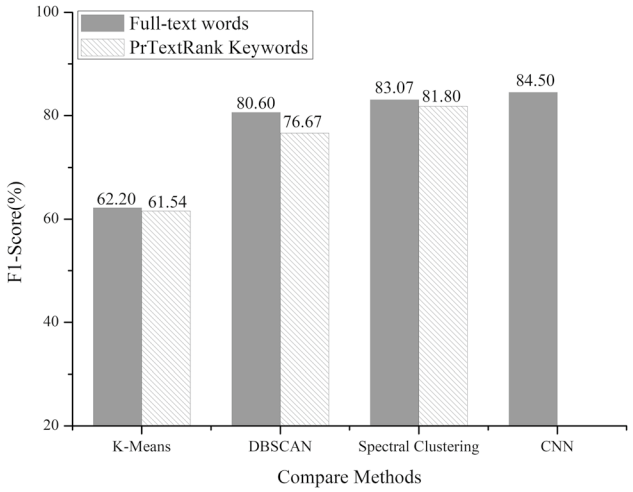

Fig. 4 F1-Score performance results for validity analysis of keywords clustering

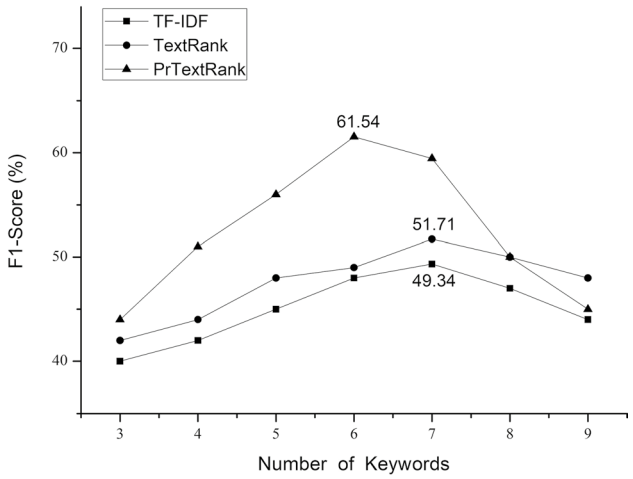

Fig. 5 F1-Score performance results under different keywords numbers

The construction of our prior knowledge network is based on the Chinese Dictionary [37], which contains items covering various fields and their explanatory information. When constructing a prior knowledge network, the interpretation words are preprocessed by word segmentation and stop words removal, and then the network is constructed using the relationship between entry items and their interpretation words.

Experimental method

Algorithm settings

In order to examine the performance of our proposed method, we use extracted keywords to represent patent texts and to perform cluster analysis. If the extracted keywords can well represent patent texts, then high-quality clustering results will be obtained. Three clustering methods, K-means [27], DBSCAN [28] and classical spectral clustering [29] are used for performance analysis.

In our experiments, the sliding window size is set to 7, the iterative time of computing rank value is set to 50, the damping coefficient is set to 0.85 by default, and the number of TopicalPageRank topics is 5. Because the number of categories of dataset I is 3, the number of categories of K-means is set to 3. Spectral clustering uses k-nearest neighbor to represent the similarity matrix, n_neighbors is set to 100, and the final number of clusters is 3. In DBSCAN clustering, ϵ is 1 and MinPts is 4.

Table 1 K-means clustering performance under different algorithm settings

	P (%)	*R* (%)	*F*1 (%)
TextRank	51.82	51.60	51.71
TextRank + $pi(v_i, D_j)$	57.98	53.40	55.59
TextRank + $W_{nn'}$	58.40	58.33	58.35
PrTextRank	**62.29**	**61.80**	**61.54**

The best results are highlighted in bold

Table 2 DBSCAN clustering performance under different algorithm settings

	P (%)	*R* (%)	*F*1 (%)
TextRank	60.49	64.60	62.48
TextRank + $pi(v_i, D_j)$	66.75	67.58	67.16
TextRank + $W_{nn'}$	71.42	72.18	71.82
PrTextRank	**77.79**	**75.56**	**76.67**

The best results are highlighted in bold

Table 3 Spectral clustering performance under different algorithm settings

	P (%)	*R* (%)	*F*1 (%)
TextRank	66.39	60.12	63.10
TextRank + $pi(v_i, D_j)$	72.18	66.60	67.58
TextRank + $W_{nn'}$	80.13	74.60	77.27
PrTextRank	**80.74**	**83.07**	**81.80**

The best results are highlighted in bold

Evaluation indices

For performance evaluation, three common evaluation indices are used including Precision (P), Recall (R) and F1-Score(F1) [38]. Corresponding calculations can be written as follows:

$$P = \frac{1}{k}\sum_{i=1}^{k}\frac{TP_i}{TP_i + FP_i} \tag{9}$$

$$R = \frac{1}{k}\sum_{i=1}^{k}\frac{TP_i}{TP_i + FN_i} \tag{10}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{11}$$

wherein, k represents the number of clusters,TP_i,FN_i,FP_i , represent the number of true positive, false negative, and false positive of category i, respectively. $TP_i + FP_i$ represents the number of samples predicted to be positive, and $TP_i + FN_i$ represents the actual positive sample number.

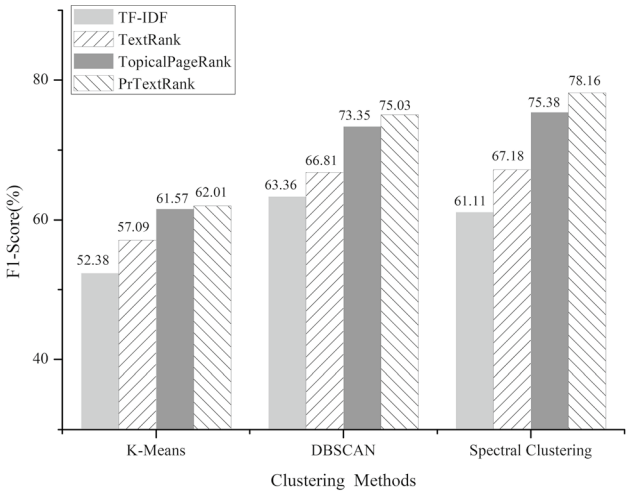

Fig. 6 Clustering performance results obtained by different methods on dataset II

Results and analysis

Here, the clustering performance respectively using full-text words and keywords are compared. Next, the performance of PrTextRank with different algorithm settings is examined. Then, the performances of PrRankText are compared with related methods. Finally, keywords extraction performance results of compared methods on non-patent documents (dataset II) are reported.

Validity analysis of keywords clustering

To verify the validity of using keywords for text clustering, we use full-text words and PrTextRank keywords as different experimental conditions, wherein three clustering algorithms are considered. In addition, baseline Convolution neural network (CNN) mentioned in [39] is also considered as comparison method. For CNN, 3000 patent texts in dataset I are divided into training data and test data with 80% and 20%. For CNN method, 12,100 features are extracted from training and testing data as CNN input, and each feature is processed as a 200-dimensional vector. In our experiments, the number of filters is set to 128, the filter window size is 3, and the step size is 1 for convolution. Max-polling is selected in the pooling layer, and the largest feature item is reserved. To prevent over-fitting, set the dropout rate to 0.5. The vector dimension is compressed by adding a flatten layer. Finally, through two dense layers, the vector length is shrunk to 3 related to three types of patent texts. Corresponding results are shown in Fig. 4.

As shown in Fig. 4, the clustering performance using keywords is slightly lower than that using full-text words. However, above performance loss seems to be very small. Besides, the performance result obtained by CNN classification reaches the maximum value 84.5%, which is 2.7% higher than the maximum value obtained by PrTextRank keywords method with spectral clustering. However, CNN method needs pre-training and is a supervised method, while PrTextRank is an unsupervised method.

Therefore, we can think that effective text clustering can be performed by using few keywords as representative of whole patent text.

Performance analysis with different keywords number

Here, we use k-means clustering to compare the clustering performance under different number of keywords. The performance results are shown in Fig. 5. Because only the title, abstract and claims are used in our experimental patent

Table 4 Performance comparison results of several related methods

	K-means			DBSCAN			Spectral clustering		
	P (%)	*R* (%)	*F*1 (%)	*P* (%)	*R* (%)	*F*1 (%)	*P* (%)	*R* (%)	*F*1 (%)
TF-IDF	48.59	50.10	49.34	57.99	58.27	57.84	54.40	69.36	60.98
TextRank	51.82	51.60	51.71	60.49	64.60	62.48	66.39	60.12	63.10
TPR	56.82	56.90	55.66	70.05	68.80	69.50	79.77	76.70	78.20
PrTextRank	**62.29**	**61.80**	**61.54**	**77.79**	**75.56**	**76.67**	**80.74**	**83.07**	**81.80**

The best results are highlighted in bold

documents, so we set the number range of extracted keywords is from 3 to 9.

From Fig. 5, PrTextRank gains the best performance when the number of keywords is 6. When the number increases from 1 to 6, the performance will also increase, while the performance will decrease when the number becomes further bigger. For above results, we may think that, more keywords will be redundant for representing patent text, and result in the decline of clustering performance. In contrast, the best keyword number for TextRank and TF-IDF is 7 according to Fig. 5. In addition, PrTextRank obtains higher performance results than two compared methods.

Performance influence under different algorithm settings

Here, Tables 1, 2 and 3 show the driving ability of different feature settings in PrTextRank under different clustering methods. Wherein, the basic model is set as TextRank, and TextRank + $P_i(V_i, D_j)$ indicates the prior importance influence, further TextRank + W_n, indicates the improved transfer factor influence.

Above experimental results show that, two newly designed algorithm strategies are very effective. Specifically, after adding prior importance of nodes in the classical TextRank model, the precision values can be increased by 6.16%, 6.26% and 5.79%, respectively. The recall values can be increased by 1.8%, 2.98% and 6.48%. The F1-Score can be increased by 3.88%, 4.68% and 4.48%, respectively. According to above results, we may believe that, effective patent text keywords may refer to more professional but not common words, and our previous considerations should be reasonable introduced in PrTextRank model.

Furthermore, it can be found that, the effective of newly designed transfer factor should be higher than that of the prior importance of nodes. These results might be consistent with the inherent idea of TextRank, that is, the importance of a node depends on the contribution of adjacent nodes, and the contribution mainly depends on the edge weight between two adjacent nodes.

Finally, when two aspects of strategies (prior node importance and improved transfer factor) in prior knowledge network are used together, the clustering performance can be further improved. Above results demonstrate that, two new considerations in PrTextRank are reasonable and effective.

Performance analysis with related methods

Here, further performances analysis of PrTextRank are performed using TF-IDF, classical TextRank and TopicalPageRank (shorten as TPR). Corresponding results are given in Table 4, in which the average results of 10 runs are taken.

For F1-Score performance index, TextRank method is 2.37%, 4.64% and 2.12% higher than TF-IDF, respectively. And, the performance of TopicalPageRank is clearly better than that of TextRank. Especially, under spectral clustering method, the F1-score obtained by TopicalPageRank is 17.22% and 15.1% higher than TF-IDF and TextRank, respectively.

However, the F1-score performances of PrTextRank are further higher than that of TopicalPageRank with 5.88%, 7.87% and 3.6% under three different clustering algorithms, respectively. For above results, the reason may be that, TopicalPageRank can cover most of text topics, however, a patent text has fewer topics and more professional term words, so using prior knowledge can effectively enhance the weight of professional term words.

Complexity analysis

In PrTextRank, a prior knowledge network should be extra constructed compared to traditional TextRank, which may increase the computational complexity. So, the running complexity of PrTextRank and TextRank is further examined on dataset I. For a case with a document containing 51 words, the running time and memory usage of TextRank method are 0.38 s and 580.0 KB. And for PrTextRank method, corresponding values are 1.26 s and 1008.0 KB. For another case with 3000 documents containing 12,100 words, the running time and memory usage of TextRank method are 990.21 s and 79,376.0 KB, while for PrTextRank method, corresponding values are 3113.59 s and 188,280.0 KB. According to above results, we may know PrTextRank requires some extra running time and running memory when constructing and using prior knowledge network. However, considering the obvious improvement in performance, those extra computational expense should be worth it.

Extended experimental analysis

To further verify the performance of PrTextRank, dataset II is used for general text keywords extraction problem. Corresponding results are shown in Fig. 6.

The results in Fig. 6 show that, PrTextRank also achieved good performance in dataset II. Compared with TF-IDF and TextRank, the performance of PrTextRank get

the improvement with 4.92%, 8.16% and 10.98%, respectively. Compared with TopicalPageRank method, the performance of PrTextRank also can improve with 0.44%, 1.68% and 2.78%, respectively. These results are consistent to the experimental results on patent text datasets. In addition, because dataset II contains sports news texts and food popular texts with completely non-overlapping topics, keywords extracted by TopicalPageRank may effectively distinguish different texts. Even so, our proposed method also gains better performance results than TopicalPageRank, which reflect the good effectiveness of introducing prior public knowledge network.

Extended analysis with latent semantic index

Keyword-based document retrieval is the simplest and most common method. However, for many documents with complex structures and content, it seems insufficient of only relying on keyword matching for the requirements of document index. So, we should also pay attention to latent semantic information in documents. The Latent Semantic Indexing (LSI) proposed by Dumais et al. [40] is powerful tool to realize document index by finding latent correlation between document words. In more specific applications, LSI may be used for complex document index task based on the extracted keywords by PrTextRank.

Conclusion and future work

In this study, we proposed a new unsupervised keywords extraction method by introducing prior public knowledge in traditional TextRank model. Wherein, two aspects of prior information are introduced including prior node importance and improved transfer factor computing strategy. Our experimental results indicate that the proposed method can obtain better performance than traditional methods on patent text keywords extraction problems. In addition, the proposed method does not need additional parameter setting, and can be widely used in different applications.

For future work, the knowledge representation of words and network construction methods could be further studied. And we also plan to explore how to integrate more multi-dimensional heterogeneous information for keywords rank value evaluation.

Acknowledgements This work was supported in part by the grants from NSFC of China (Grant no. 61872166) and Six Talent Peaks Project of Jiangsu Province of China (Grant no. 2019 XYDXX-161)

Author contributions Proposes a graph-based patent keyword extraction algorithm by means of a prior public knowledge, and demonstrates the effectiveness of the algorithm by adequate experimental analysis.

Funding This work was supported in part by the grants from NSFC of China (Grant no. 61872166) and Six Talent Peaks Project of Jiangsu Province of China (Grant no. 2019 XYDXX-161).

Availability of data and material The data used to support the findings of this study will be available from the corresponding authors.

Code availability Custom code

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Joung J, Kim K (2017) Monitoring emerging technologies for technology planning using technical keyword based analysis from patent data. Technol Forecast Soc Chang 114:281–292. https://doi.org/10.1016/j.techfore.2016.08.020
2. Li Y-R, Wang L-H, Hong C-F (2009) Extracting the significant-rare keywords for patent analysis. Expert Syst Appl 36:5200–5204. https://doi.org/10.1016/j.eswa.2008.06.131
3. Hernández-Castañeda Á, García Hernández RA, Ledeneva Y, Millán-Hernández CE (2020) Extractive automatic text summarization based on lexical-semantic keywords. IEEE Access 8:49896–49907. https://doi.org/10.1109/ACCESS.2020.2980226
4. Gerken JM, Moehrle MG (2012) A new instrument for technology monitoring: novelty in patents measured by semantic patent analysis. Scientometrics 91:645–670. https://doi.org/10.1007/s11192-012-0635-7
5. Jin CX, Zhou HY, Bai QC (2012) Short text clustering algorithm with feature keyword expansion. In: Materials science and information technology II. Trans Tech Publications Ltd, pp 1716–1720
6. Onan A, Korukoğlu S, Bulut H (2016) Ensemble of keyword extraction methods and classifiers in text classification. Expert Syst Appl 57:232–247. https://doi.org/10.1016/j.eswa.2016.03.045
7. Mihalcea R, Tarau P (2004) TextRank: bringing order into texts. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP).ACL, Stroudsburg, Pennsylvania, pp. 404–411.

8. Ding W, Wang J, Zhu H (2019) Using graph representations for semantic information extraction from chinese patents. In: Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control. ACM, New York, NY, USA, pp. 1–5.
9. Page L, Brin S, Motwani R, Winograd T (1999) The pagerank citation ranking: bringing order to the web. Technical Report, Stanford InfoLab
10. Ardiansyah S, Majid M, Mohamad Zain J (2016) Knowledge of extraction from trained neural network by using decision tree. In: Proceedings of the International Conference on Science in Information Technology (ICSITech). IEEE, Balikpapan, Indonesia, pp 220–225.
11. Sangeetha J, Jothilakshmi S (2014) A novel spoken keyword spotting system using support vector machine. Eng Appl Artif Intell 36:287–293. https://doi.org/10.1016/j.engappai.2014.07.014
12. Wang J, Song F, Walia K, et al (2019) Using convolutional neural networks to extract keywords and keyphrases: a case study for foodborne illnesses. In: Proceedings of the 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, Florida, , pp 1398–1403.
13. Zhang Y, Tuo M, Yin Q et al (2020) Keywords extraction with deep neural network model. Neurocomputing 383:113–121. https://doi.org/10.1016/j.neucom.2019.11.083
14. She C, You H, Lin C, et al (2020) Deep neural semantic network for keywords extraction on short text. In: Qin P, Wang H, Sun G, Lu Z (eds) Data Science. ICPCSEE 2020. Springer, Singapore, pp 101–112.
15. Feng Y, Zhang H, Hao W, Chen G (2017) Joint extraction of entities and relations using reinforcement learning and deep learning. In: Computational Intelligence and Neuroscience. https://www.hindawi.com/journals/cin/2017/7643065/. Accessed 23 Nov 2020.
16. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inf Process Manage 24:513–523. https://doi.org/10.1016/0306-4573(88)90021-0
17. Gollapalli SD, Caragea C (2014) Extracting keyphrases from research papers using citation networks. In: Proceedings of the Twenty-eighth AAAI Conference on Artificial Intelligence (AAAI-14). AAAI Press, Québec, Canada, pp 1629–1635.
18. Florescu C, Caragea C (2017) PositionRank: an unsupervised approach to keyphrase extraction from scholarly documents. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2017).ACL, Vancouver, Canada, pp 1105–1115.
19. Devika R, Subramaniyaswamy V (2019) A semantic graph-based keyword extraction model using ranking method on big social data. Wireless Netw. https://doi.org/10.1007/s11276-019-02128-x
20. Liu Z, Huang W, Zheng Y, Sun M (2010) Automatic keyphrase extraction via topic decomposition. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Cambridge, Massachusetts, pp 366–376.
21. Yang SY, Lin SY, Lin SN, et al (2008) Automatic extraction of semantic relations from patent claims. Int J Electron Bus Manag.
22. Noh H, Jo Y, Lee S (2015) Keyword selection and processing strategy for applying text mining to patent analysis. Expert Syst Appl 42:4348–4360. https://doi.org/10.1016/j.eswa.2015.01.050
23. Madani F, Weber C (2016) The evolution of patent mining: applying bibliometrics analysis and keyword network analysis. World Patent Inf 46:32–48. https://doi.org/10.1016/j.wpi.2016.05.008
24. Park Y, Yoon J (2017) Application technology opportunity discovery from technology portfolios: Use of patent classification and collaborative filtering. Technol Forecast Soc Chang 118:170–183. https://doi.org/10.1016/j.techfore.2017.02.018
25. Yanagihori K, Tsuda K (2013) Issues of the morphological analysis in comparison with the compound noun extraction analysis for a patent document. Inform Syst Int Conf 2013.
26. Sinaga KP, Yang M (2020) Unsupervised K-means clustering algorithm. IEEE Access 8:80716–80727. https://doi.org/10.1109/ACCESS.2020.2988796
27. Bide P, Shedge R (2015) Improved document clustering using k-means algorithm. In: Proceedings of the 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE, Coimbatore, India, pp 1–5.
28. Chen S, Liu X, Ma J et al (2019) Parameter selection algorithm of DBSCAN based on K-means two classification algorithm. J Eng 2019:8676–8679. https://doi.org/10.1049/joe.2018.9082
29. Nataliani Y, Yang M-S (2107) Powered gaussian kernel spectral clustering. Neural Comput Appl 31:1–16. https://doi.org/10.1007/s00521-017-3036-2
30. Boudin F, Mougard H, Cram D (2016) How document pre-processing affects keyphrase extraction performance. In: Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT), Osaka, Japan, pp 121–128.
31. Witten I, Paynter G, Frank E, et al (1999) KEA: practical automatic keyphrase extraction. In: Proceedings of the Fourth ACM Conference on Digital Libraries (JCDL). ACM, New York, pp. 254–255.
32. Michel AN, Farrell J (1990) Associative memories via artifical neural networks. Control Syst Mag IEEE 10:6–17. https://doi.org/10.1109/37.55118
33. Bassett DS, Mattar MG (2017) A network neuroscience of human learning: potential to inform quantitative theories of brain and behavior. Trends Cogn Sci 21:250–264. https://doi.org/10.1016/j.tics.2017.01.010
34. Xie Z, Wang K, Liu Y (2020) On learning associative relationship memory among knowledge concepts. Int J Netw Distrib Comput 8:124. https://doi.org/10.2991/ijndc.k.200515.005
35. Wang C, Zhang M, Ma S, Ru L (2008) Automatic online news issue construction in web environment. In: Proceedings of the 17th International Conference on World Wide Web. ACM, New York, NY, USA, pp 457–466.
36. Foodbk. [EB/OL]. [Online]. Available via DIALOG. http://www.foodbk.com/ of subordinate document.
37. The Chinese Dictionary. [EB/OL]. [Online]. Available via DIALOG. http://www.hydcd.com/.
38. Zhu Y, Zheng W, Tang H (2020) Interactive dual attention network for text sentiment classification. Comput Intell Neurosci 2020:8858717. https://doi.org/10.1155/2020/8858717
39. Li Q, Li P, Mao K, Lo EY-M (2020) Improving convolutional neural network for text classification by recursive data pruning. Neurocomputing 414:143–152. https://doi.org/10.1016/j.neucom.2020.07.049
40. Deerwester S, Dumais ST, Furnas GW et al (1990) Indexing by latent semantic analysis. J Am Soc Inform Sci 41:391–407