JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 35, 651-674 (2019)
DOI: 10.6688/JISE.201905_35(3).0010

Sentence-Ranking-Enhanced Keywords Extraction from Chinese Patents*

ZHI-HONG WANG[1] AND YI GUO[1,2,3,+]
[1]*Department of Computer Science and Engineering*
East China University of Science and Technology
Shanghai, 200237 P.R. China
[2]*Business Intelligence and Visualization Research Center*
National Engineering Laboratory for Big Data Distribution and Exchange Technologies
Shanghai, 200436 P.R. China
[3]*School of Information Science and Technology*
Shihezi University
Shihezi, 8320003 P.R. China

Patent keywords, a high-level topic representation of patents, hold an important position in many patent-oriented mining tasks, such as classification, retrieval and translation. However, there are few studies concentrated on keywords extraction for patents in current stage, and neither exist human-annotated gold standard datasets, especially for Chinese patents. This paper introduces a new human-annotated Chinese patent dataset and proposes a sentence-ranking based Term Frequency-Inverse Document Frequency (SR based TF-IDF) algorithm for patent keywords extraction, motivated by the thought of "the keywords are in the key sentences". In the algorithm, a sentence-ranking model is constructed to filter top-K_S percent sentences from each patent based on a sentence semantic graph and heuristic rules. At last, the proposed algorithm is evaluated with TF-IDF, TextRank, word2vec weighted TextRank and Patent Keyword Extraction Algorithm (PKEA) on the homemade Chinese patent dataset and several standard benchmark datasets. The experimental results testify that our proposed algorithm effectively improves the performance of extracting keywords from Chinese patents.

Keywords: Chinese patents, key sentences, sentence-ranking model, keywords extraction, human-annotated dataset

1. INTRODUCTION

Patent is an important category of scientific literature, which contains academic, commercial, legal and some other information. It records a large amount of innovative discoveries and practical research conclusions originated from industry and academia. In another words, the trend of new technologies can be speculated and driven towards new applications with analysis of patent bibliography, changes in patent legal status or citation relations [1]. The comprehensive analysis of patents becomes one of the most important measures for assessing the technological competitive power of nations. To be noted, the number of Chinese patents has accumulated to 40,673,532 by October 2017

Received December 2, 2017; revised May 13, 2018; accepted July 8, 2018.
Communicated by Berlin Chen.
* This research is financially supported by National Key Research and Development Program of China Grant No. 2018YFC0807105, National Natural Science Foundation of China Grant No. 61462073 and Science and Technology Committee of Shanghai Municipality (STCSM) Grant Nos. 17DZ1101003, 18511106602 and 18DZ2252300.
+ Corresponding author: guoyi@ecust.edu.cn

according to the latest announcements of State Intellectual Property Office (SIPO) [2], up to a third of the total global patents (Table 1 states the distribution of global patents.). Meanwhile, the number of Chinese patents still keeps high annual growing rates [3]. For example, the invention patents increased with a rate of approximately 9% last year. All in all, it is of great importance to analyze and investigate the potential value from massive Chinese patents.

Table 1. The distribution of global patents.

Countries /Regions	Time ranges	Data size	Countries /Regions	Time ranges	Data size
CN	1985.09.10~2017.10.03	40,673,532	US	1790.07.31~2017.09.14	15,786,227
JP	1913.02.06~2017.09.11	39,893,753	KR	1973.10.23~2017.08.31	4,591,866
GB	1782.07.04~2017.09.20	3,687,038	FR	1855.02.27~2017.09.15	3,122,114
DE	1877.07.02~2017.09.21	7,553,081	RU	1992.10.15~2017.09.08	1,230,382
CH	1888.01.09~2017.08.15	724,431	EP	1978.12.20~2017.09.20	5,973,129
WO	1978.10.19~2017.09.14	4,649,505	Others	1790.07.31~2017.09.14	15,432,509

Many patent-oriented mining tasks, such as classification, retrieval and translation, rely on the extraction of representative keywords in a large extent. Wu *et al.* [4] constructed a Weighted Keyword-based Patent Network (WKPN) approach to estimate technological trends and evolution of biofuels in a green energy field. Joung *et al.* [5] proposed a technical keyword-based approach to trace emerging technologies based on TF-IDF. Patent keywords have been widely applied in patent mining. As is known to all, keywords extraction algorithms achieved great progress in last decades. However, the current keywords extraction algorithms target on texts, such as news articles [9, 16, 18], emails [22], scientific papers [12-16], technical reports [10] or Web pages [17], while the patent documents have several distinctive features. For example, the unregistered terms appear frequently in patents, and with rapid development of new technologies, new terms come forth constantly, such as "Long Short-Term Memory" (LSTM), "Gated Recurrent Unit" (GRU). Moreover, patents contain some domain-dependent technical terms, which present only in specific fields but rare in others. In addition, the technical statements are often reiterated in contents, and the terms or expressions in patents are rigorous and followed much certain criteria imposed by intellectual property organizations. Therefore, the current research works on patent keywords extraction are still in absence due to the following issues, such as unregistered terms, synonyms and repeated expressions. Besides, all patents are not accompanied, in nature, with author-assigned keywords, which create a huge challenge for manual keywords assignment for each patent.

This paper proposes a new approach to extract patent keywords based on a sentence-ranking model. The key insight is that the keywords are in the key sentences, inspired by the reversing idea of text summarization. In addition, a new human-annotated Chinese patent dataset is submitted for the task of keywords extraction. We attempt to address the solution of automatic keywords extraction from Chinese patents and the contributions are concluded as follows.

- Produce an annotated dataset with keywords, consisting of 839 Chinese patents from SIPO and each has 3 to 6 manual annotated keywords. The dataset is accessible in our

google drive[1].
- Construct a sentence-ranking model based on a sentence embedding graph and heuristic rules.
- Propose a Sentence-Ranking based Term Frequency and Inverse Document Frequency (SR based TF-IDF) algorithm for keywords extraction from Chinese patents, inspired by the thought of "the keywords are in the key sentences".
- Evaluate SR based TF-IDF with the most commonly used and the latest keywords extraction algorithms, including TF-IDF, TextRank, word2vec based TextRank (2017) and PKEA (2018).

The rest of this paper is organized as follows. Section 2 describes the closely related works. Section 3 details the architecture of our keywords extraction system (SR based TF-IDF). Section 4 evaluates our models (SR based TF-IDF) with dedicated experiments and Section 5 concludes this paper.

2. RELATED WORK

2.1 Keywords Extraction Pipeline

Keywords extraction is usually implemented in four phases. Pre-Processing is commonly used as a preliminary practice, which includes sampling, transformation, denoising, or segmentation. Candidate Selection identifies and selects words/phrases of potential importance from the texts using some heuristic rules. Keywords Ranking/Classification scores keywords with several supervised or unsupervised algorithms from the candidate words, and Post-Processing merges the neighboring words into a single readable phrase from the top-*k* words.

Pre-Processing: In this phase, the title and text are extracted based on special heuristic rules [6] or text extraction algorithms [7]. At the same time, a long text should be segmented into several paragraphs with paragraph marks (carriage return character, line feeds *etc.*), and paragraphs are required to be segmented into several sentences with punctuation [6]. In addition, Chinese texts have no such a clear demarcation between words like English. Some basic operations therefore are required in the pre-processing phase, such as word segmentation, part of speech tagging, new word detection [9].

Candidate Selection: This phase is to determine the keywords candidate collection, which will improve the efficiency of keywords extraction. Heuristic rules of high quality are practiced in candidate words selection. For example, a limited maximum/minimum length [10], stop words removal [11, 13], *n*-gram phrases [11-13], words with proper co-occurrence probabilities [12], certain part-of-speech (POS) tags (noun, verb, *etc.*) [14]. Many of these heuristics are proven effective with their high recall in extracting gold key phrases from various sources. However, candidate keywords are still with a wide range and contain many non-grammatical phrases after selecting with the above rules [11].

Keywords Ranking/Classification: It is hard to choose an effective extraction algorithm

[1] Human-annotated Patents: https://drive.google.com/open?id=1jgzE19rLgLR2sE4OIFqltQyhAbzzCuvm

(ranking or classification) to determine which of these candidates are correct keywords. In these algorithms, the most known is graph-ranking-based algorithm. TextRank [15] is one of the most well-known graph-ranking-based approaches for keywords extraction. But the word graph generated in TextRank doesn't measure word importance with respect to different topics. Liu *et al.* [16] proposed Topical PageRank (TPR) to run TextRank once for each topic induced by a Latent Dirichlet Allocation (LDA). In addition, word graph is built by co-occurrence in sliding window, the effect of the approach is obviously not good for short text and the weights between two words without any semantic information. Therefore, some researchers introduce external knowledge base, such as Wikipedia [17], to enrich short text information for keywords extraction. Moreover, some others propose different methods to calculate the weights between the graph words, such as word2vec [18], Collective Node Weight [19]. The basic idea behind these approaches is to build a graph with the candidate words and rank them using a graph-based ranking method.

Besides, the classification algorithms are also efficient in this phase even have better performance in some fields. In a study by Hasan and Ng [20], TF-IDF is proven as a surprisingly robust candidate and beats other more complex ranking strategies. Supervised classification approaches are mainly focused on feature design. Other important features for classification methods include *tfidf* [20], *phrase length* [10, 22], *word position* [14, 22], *word diameter* [21], *POS tags* [22] and *is-in-Wikipedia* [23] *etc.*

Post-Processing: The final important phase in keywords extraction is post-filtering, such as filtering short words, limiting the number of two Chinese words [20]. And adjacent words are also sometimes collapsed into phrases, for a more readable output.

2.2 Keywords Extraction in Patents

Research works on patents mainly focus on keywords-based patent mining include technology evolution analysis, future technological trends analysis, patent translation, patent retrieval and patent classification *etc.* Chen and Zhang [24] proposed a topic-based prediction approach to identify trends in technology, and proved the effectiveness in future trends prediction based on 13,910 patents published in Australia between 2000 and 2014. Hu *et al.* [25] presented a patent keyword extraction algorithm (PKEA) based on the distributed Skip-gram model for patent classification. Many studies have developed methodologies based on patent keywords to analyze the patents in quantitative manner so that experts can read and analyze these studies using text mining. Therefore, keywords extraction in patents makes a great difference to the patent analysis.

The situation of Chinese patents is invariable, the studies primarily focus on technological competitive analysis about enterprises, industries or regions based on keywords from Chinese patents [26]. Ding *et al.* [27] linked all patents by keywords to improve the speed of patent retrieval. Moreover, patents were also utilized to as a background knowledge base to realize a better keywords automatic extraction algorithm in other fields [28]. Liu and Peng [29] proposed a semi-automatic patented-technical phrase extraction method, which effectively reduced labor cost and achieved good results on Chinese patents.

Based upon above reviews, there are not enough research works conducted in keywords extraction from patents, especially for Chinese. Due to the specialty and unique-

ness of patents, traditional keywords extraction algorithms cannot be applied to Chinese patents in a direct manner. Therefore, this paper proposes a new patent keywords extraction algorithm based on a sentence-ranking model (SR based TF-IDF). The experimental results testify that our algorithm outperforms the most commonly used TF-IDF, TextRank and the latest word2vec weighted TextRank (2017), PKEA (2018).

3. AUTOMATIC KEYWORDS EXTRACTION FROM CHINESE PATENTS

3.1 Overall Research Framework

From the above discussion, a research framework, consists of a domain dictionary construction (DDC) module and a keywords extraction model (KEM), is proposed in this paper for keywords extraction from Chinese patents (Fig. 1).

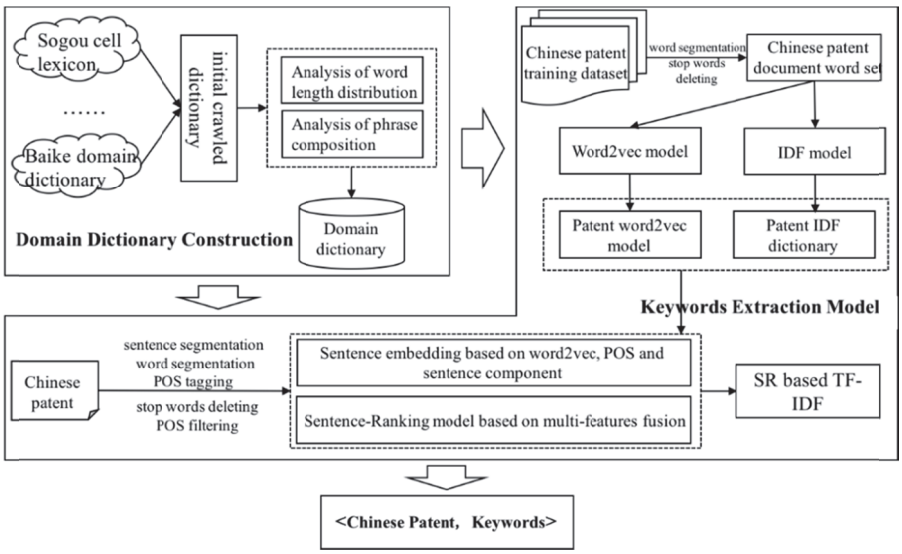

Fig. 1. The overall framework of keywords extraction from Chinese patents.

In Fig. 1, the DDC module constructs a domain dictionary for unregistered words recognition from Chinese patents, including initial lexicons collection, phrase length and composition analysis (See Section 3.2). The next KEM module includes sentence embedding model (See Section 3.3.1), sentence-ranking model (See Section 3.3.2) and SR based TF-IDF (See Section 3.3.3). In the end, the system produces a fixed number of keywords for each Chinese patent.

3.2 Domain Dictionary Construction

In general, keywords extraction for Chinese texts achieves better results with the high-quality word segmentation. Several methods based on external dictionaries [28] or new word detection [30] are proven effective with their high accuracy in extracting key-

words. As mentioned above, Chinese patents contain a large number of professional terms. To better improve the performance of keywords extraction, a domain dictionary is constructed by merging multiple source lexicons, including the 861 lexicons under the "Engineering Application" of Sogou cell lexicon [31] and all entries under the science category of Baidu Baike [32]. At last, the dictionary contains about 130 million words (or phrases) in total. However, there are several problems in this dictionary, which are listed as follows:

- Duplicate entries with the same meaning caused by uppercase and lowercase or Simplified-Chinese and Traditional-Chinese.
- Meaningless long entries, such as "线路由运行转为冷备用" (the line transferred from running to cold standby).
- Combined-entries, such as "不易燃和无毒" (non-flammable and non-toxic)

This paper tackles above issues in following steps to produce a higher quality domain dictionary. The first step is to convert words/phrases of the dictionary into the normalized form (English in lowercase and Chinese in simplified) and delete the redundancy. The distribution of the word length is listed in Table 2, which notes that the shortest word length in the dictionary is one Chinese character and the longest one is 43 characters. To ensure the quality and coverage, the dictionary keeps all entries with the length from 2 to 7, which is accounted for 0.906. At last, the word segmentation and POS tagging are implemented for each word/phrase by a Natural Language Processing (NLP) tool, such as LTP [33]. According to the combination rules of Chinese words [34], words/phrases generally do not contain conjunction (such as "和" (and)), preposition (such as "在" (in)), auxiliary (such as "是" (is)), adverb (such as "很" (very)), and punctuation (such as "."). Thus, the words/phrases, which contain these ban-words, are all deleted. 284,328 words are obtained eventually.

Table 2. The distribution of word length.

Word length	Proportion	Word length	Proportion
2	0.117	8	0.034
3	0.170	9	0.019
4	0.282	10	0.012
5	0.155	11	0.006
6	0.114	12	0.005
7	0.068	Others (1,13-43)	0.018

3.3 Sentence-Ranking Based Keywords Extraction Algorithm

Extractive text summarization is an effective way to reduce a text into a summary by selecting, in an automatic manner, a subset of the text [35]. The central idea of these methods is to seek the most commonly used phrases (such as key-phrases/ keywords) that encompass the topic sentence. Afterwards, the sentence is treated as a part of the summary [36]. That is to say 'a sentence' which contains more frequent words (such as key-phrases/keywords), is more important than other sentences. Correspondingly, a word, appears frequently in the important sentence, is more likely to be the candidate keywords

of the document. Inspired by the simple observation, this paper proposes a new keyword extraction algorithm based on a sentence-ranking model.

The sentence-ranking model integrates the sentence graph and heuristic rules. More specifically, the sentence graph is built to sort the sentences by a graph-based ranking method, such as PageRank. Zhang *et al.* [37] built a sentence graph, which treats each sentence as a vertex, the sentence relations as the edges and Jaccard similarity as the weight of each edge. However, the similarity in [37] was computed only with the overlapped words between sentences without consideration of the potential semantic information among them. Therefore, a patent sentence embedding model is introduced in Section 3.3.1 to describe the semantic similarity between sentences in a more accurate way. On this basis, a sentence-ranking model (see Section 3.3.2) is constructed to select top-K_S percent sentences. In the end, the top-K_W keywords are extracted with an improved TF-IDF algorithm (see Section 3.3.3) from the selected sentences.

3.3.1 Patent sentence embedding

Much progress has been achieved in learning semantically meaningful distributed representations of individual words, also known as word embeddings. On the other hand, several researches also attempt to obtain satisfying phrase or sentence representations. However, most of the methods are proposed especially for a certain task [38], such as Information Retrieval (IR) [39], Question Answering (QA) [40]. To our knowledge, there are no sentence representations for patent sentences, especially for the task of patent keywords extraction. Therefore, a sentence embedding model for patents is proposed based on word2vec [41] and heuristic rules to represent the potential semantic similarity between patent sentences.

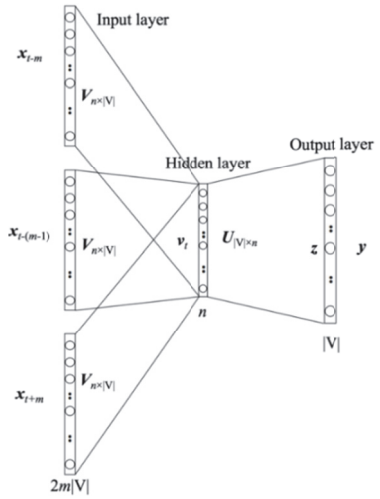

Fig. 2. The architecture of CBOW model [42].

Word2vec takes a large corpus of texts as its input and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being as-

signed a corresponding vector in the space. We employ the continuous bag-of-word (CBOW) model as our distributed word representation approach for word2vec. The CBOW model takes the average of the vectors of the input context words, and uses the product of the input→hidden weight matrix and the average vector as the output. Fig. 2 shows the architecture of the CBOW model. In the training process of this algorithm, the hyper-parameters are set as follows: The minimum word count (ignore all words with total frequency lower than this), window size (the maximum distance between the current and predicted word within a sentence) and the dimensionality of the feature vectors for patent words embedding is set to 5, 5 and 400 respectively.

Besides, the features of words and sentences used are as follows:

- Sentence is a collection of words;
- Words with different POS tags have different contributions to the sentences;
- Words in different sentence components have different contributions to the sentences.

Thus, for a sentence S, $SW = (sw_1, sw_2, \ldots, sw_m)$, where sw_i ($i = 1, 2, \ldots, m$) is the ith word in sentence S. The corresponding POS of SW is denoted as $SN = (sn_1, sn_2, \ldots, sn_m)$, where sn_i ($i = 1, 2, \ldots, m$) represents the POS of the corresponding word sw_i ($i = 1, 2, \ldots, m$). The word sw_i ($i = 1, 2, \ldots, m$) in SW can be represented by a k*1 vector based on word2vec, named $V_{sw_i} = [v_{i1}, v_{i2}, \ldots, v_{ik}]^T$, where $v_{ij} \in \mathbb{R}$, $j = 1, 2, \ldots, k$. Thus, the sentence embedding model is defined as:

$$V_S = \sum_{i=1}^{m} (w_{pos} + w_{sc}) * V_{sw_i} \tag{1}$$

where V_S is the sentence embedding of sentence S and m is the total number of words in sentence S. sw_i represents the ith word in sentence S and V_{sw_i} is its word embedding calculated by word2vec. w_{pos} is the weight of candidate keywords with different POS tags. The sentence embedding model utilizes the analysis results of POS tags distribution of keywords from the study of Zhang [43], and $w_{pos} = \begin{cases} 0.8, & \text{if } sn_i \text{ is noun} \\ 0.5, & \text{if } sn_i \text{ is verb} \\ 0.4, & \text{if } sn_i \text{ is adj} \\ 0, & \text{if } sn_i \text{ is others} \end{cases}$. w_{sc} is the weight of candidate keyword in different sentence components, such as subject, predicate, object or others, and the weights are from [44], and $w_{sc} = \begin{cases} 0.5, & \text{if } sw_i \text{ is subject} \\ 0.2, & \text{if } sw_i \text{ is predicate} \\ 0.3, & \text{if } sw_i \text{ is object} \\ 0, & \text{if } sw_i \text{ is others} \end{cases}$.

3.3.2 Multi-feature fusion based sentence-ranking model

This section presents the details of the sentence-ranking model based on Heuristic Rules (See Section 3.3.2 (A)) and a Sentence Semantic Graph (See Section 3.3.2 (B)), which will be described more at length. For convenience, some symbolic variables are given in first. Suppose that for a Chinese patent document P, the title is T. The abstract

sentences in P are S, and $|S| = n$. The goal of sentence-ranking model is to compute an n-dimensional vector $SR = [SR_1, SR_2, \ldots, SR_n]^T$, where SR_i is the weight of the ith sentence. Hence, the top-K_S percent sentences with the highest score can be achieved from SR.

(A) Heuristic Rules

Patent, a kind of scientific literature, has strict standardized templates and writing criterions. More specially, the title of patent indicates the subject and type of a patent; the abstract gives a brief technical summary of the fields, issues and characteristics. According to the analysis and our observation of patents, the following heuristic rules are taken into consideration:

- The more similarity with title the sentence is, the more important the sentence is.
- The sentence in different position is with different importance. Generally, the first and the last sentence are more important than others.

Suppose that S_i is the ith sentence in the set S of patent abstract sentences, and that $SW_i = (sw_{i1}, sw_{i2}, \ldots, sw_{im})$ represents all words in the sentence S_i, where sw_{ij} ($j = 1, 2, \ldots, n$) is the jth word in the sentence S_i. All words in patent title T are represented by $TW = (tw_1, tw_2, \ldots, tw_t)$, where tw_i ($i = 1, 2, \ldots, t$) is the ith word in title T. Thus, the similarity between the patent title T and the sentences in patent abstract S is shown in Eq. (2).

$$W_{TitleOverlap}(S, T) = [to_1, to_2, \ldots, to_n]^T \tag{2}$$

where to_i is the similarity between the ith sentence in patent abstract with the patent title T, which is calculated by Jaccard similarity in Eq. (3).

$$to_i = Jaccard(S_i, T) = \frac{|SW_i \cap TW|}{|SW_i \cup TW|} \tag{3}$$

Meanwhile, the weight of different position of patent abstract sentences is defined in Eq. (4).

$$W_{location}(S) = [loc_1, loc_2, \ldots, loc_n]^T \tag{4}$$

where loc_i is the location weight of the ith abstract sentence in S. According to the sampling statistics of P. E. Baxendale [45], 85% of the sentences, reflect the theme of the document, appears at the beginning of the paragraph, and 7% is in the end. Therefore, the location weight loc_i of sentence S_i is defined as follows.

When $n > 2$, define loc_i by

$$loc_i = \begin{cases} 0.85, & \text{if } S_i \text{ is the first sentence.} \\ 0.07, & \text{if } S_i \text{ is the last sentence.} \\ \frac{0.08}{n-2}, & \text{if } S_i \text{ is the other sentences.} \end{cases} \tag{5}$$

When $n = 2$, there are only two sentences in the patent abstract, the first sentence and the last sentence, and define loc_i by

$$loc_i = \begin{cases} 0.89, & \text{if } S_i \text{ is the first sentence} \\ 0.11, & \text{if } S_i \text{ is the last sentence} \end{cases}. \tag{6}$$

When $n = 1$, there is only one sentence in the patent abstract and define loc_i by

$$loc_i = 1. \tag{7}$$

According to Eqs. (2) and (4), the weight of patent abstract sentences based on heuristic rules proposed is defined as Eq. (8).

$$W_{rule}(S) = \alpha * W_{TitleOverlap}(S, T) + (1 - \alpha) * W_{location}(S) \tag{8}$$

Where α is the weight of one of the heuristic rules, the other is $1 - \alpha$. While $W_{TitleOverlap}(S, T)$ is calculated with Eq. (2) and $W_{location}(S)$ is calculated with Eq. (4).

(B) A Sentence Semantic Graph

Addressing the potential semantic information between sentences, a sentence semantic graph named G is built on sentence embeddings. The graph takes the sentences in patent abstract as the vertex and the similar relation between the sentences as the edges. Furthermore, PageRank is selected as the graph sorting algorithm in this study. The adjacency matrix of semantic similarity between sentences is defined in Eq. (9).

$$p_{sim(s)} = [s_{ij}]_{n \times n} \tag{9}$$

where s_{ij} is the weight of (S_i, S_j), which is defined by the semantic similarity between the ith and the jth sentence in patent abstract sentences S. Moreover, the weight is calculated by the cosine similarity based on sentence embedding as below:

$$s_{ij} = \cos(V_{S_i}, V_{S_j}) = \frac{V_{S_i} \bullet V_{S_j}}{\left\| V_{S_i} \right\| \left\| V_{S_j} \right\|} \tag{10}$$

where S_i is the ith sentence. V_{S_i} is its sentence embedding, calculated with Eq. (1).

The iterative formula based on the thought of PageRank, which is used to calculate the sentence weight on the sentence semantic graph G as below:

$$w_{PR}(S_i) = (1 - d) + d * \sum_{s_{ij} \neq 0} \frac{s_{ij}}{\sum_k s_{ik}} w_{PR}(S_j) \tag{11}$$

where d is the damping factor. And $w_{PR}(S_i)$, which can be any non-negative values at initialization, is given by the last iteration in the subsequent iterations.

Like the random walk model, the above iterative process can be converted into matrix operations. Suppose that W_{PR}^i is the weight vector of patent abstract sentences in the ith iteration then the Eq. (11) can be re-expressed as:

$$W_{PR}^i = PW_{PR}^{i-1}. \tag{12}$$

The above matrix equation gives a more concise iterative process of sentence weight calculation based on a sentence semantic graph. That is, the vector is first initialized with random values and then iteratively updated according to Eq. (12) until convergence.

In summary, according to heuristic rules and a sentence sematic graph, the sentence-ranking model for Chinese patents is defined in Eq. (13).

$$SR(S) = [SR_1, SR_2, \ldots, SR_n]^T = \beta * W_{rule}(S) + (1-\beta) * W_{PR}(S) \tag{13}$$

where, β is the weight of heuristic rules. While $W_{rule}(S)$ is calculated with Eq. (8) and $W_{PR}(S)$ is calculated with Eq. (12).

3.3.3 Sentence-ranking based keywords extraction algorithm

Since the sentences are treated equally in traditional keywords extraction algorithms without considering the semantic weights of different sentences. The semantic importance of different sentences can be determined by the sentence-ranking model. Thus, the semantic weights of each sentence are introduced into the commonly used algorithm Term Frequency (TF), so that the semantic information of sentences can be transmitted into the words. TF is defined as follows.

$$TF(w_i) = \sum_{j=1}^{K_S * n} SR_j * TF_j(w_i) \tag{14}$$

where w_i is the ith word in patent abstract. n is the total number of sentences in patent abstract. K_S*n is the number of sentences with the highest weight. SR_j is the semantic weight of the jth sentence. $TF_j(w_i)$ is the term frequency of word w_i in the jth sentence. The pseudo-code of SR based TF-IDF is shown in Algorithm 1.

Algorithm 1: SR based TF-IDF

Input: P is the adjacency matrix of semantic similarity between sentences. T is the title of patent. n is the total number of sentences in patent abstract. IDF is the inverse document frequency of patents. d, α, β, ε, K_S, K_W are the parameters of Algorithm 1.

Output: $KS<S_i, SR_i>$ is the set of key sentences, which is initial with ϕ.
KW is the set of keywords, which is also initial with ϕ.

1) Randomly initialize vector W^*_{PR}, $i \leftarrow 0$;
2) while $||W^*_{PR}-W^{*-1}_{PR}||^2 \geq \varepsilon$ do:
3) $i \leftarrow i+1$;
4) update W^*_{PR} according to Eq. (12)
5) end while
6) for each S_i in S:
7) update $W_{TitleOverlap}(S_i, T)$ according to Eq. (3);
8) if $n == 1$ then:
9) update $W_{location}(S_i)$ according to Eq. (7);
10) else if $n == 2$ then:
11) update $W_{location}(S_i)$ according to Eq. (6);
12) else:

13) update $W_{location}(S_i)$ according to Eq. (5);
14) end if
15) end for
16) $W_{rule}(S) \leftarrow \alpha * W_{TitleOverlap}(S_i) + (1 - \alpha) * W_{location}(S)$
17) $SR(S) \leftarrow \beta * W_{rule}(S) + (1 - \beta) * W_{PR}(S)$
18) while $|KS| \leq K_S * n$ do:
19) select the sentence S_i with the highest score in S according SR_i;
20) end while
21) for w_i in KS:
22) update $TF(w_i)$ according to Eq. (14);
23) $TF\text{-}IDF(w_i) \leftarrow TF(w_i) * IDF(w_i)$;
24) end for
25) while $|KW| \leq K_W$ do:
26) select the keyword w_i with the highest score according to $TF\text{-}IDF(w_i)$;
27) end while

This algorithm involves six parameters in total, among which, the damping factor d is generally set as 0.85 according to PageRank algorithm. The remaining parameters are discussed in the next Section 4.2.

4. PERFORMANCE EVALUATION

This section details the experimental datasets and results. The main goal of this section is to validate whether our proposed algorithm is effective for extracting keywords from patents. More specifically, the Section 4.1 gives a detailed description of our experimental datasets and metrics. The parameters of the proposed algorithm are discussed in Section 4.2. The experimental results carried out by SR based TF-IDF algorithm and the baseline keyword extraction algorithms are reported in Section 4.3. In the end, two additional evaluations on Chinese and English datasets further illustrate that the proposed approach is scalable and applicable in other fields or languages (See Section 4.4).

4.1 Datasets and Metrics

Dataset 1: *Inspec* [14]. This dataset consists of 2,000 abstracts in English from Inspec database, including their corresponding title and keywords. Each abstract has two sets of keywords, controlled terms that restricted to the Inspec thesaurus, and uncontrolled terms that can be any suitable terms. Only the uncontrolled terms are considered in this study.

Dataset 2: A public Chinese academic paper dataset. 1,654 Chinese academic papers about "Big Data" are crawled from Wan Fang [46], a Chinese Journal Database, and the dataset is open accessible in our google drive[2]. In addition, the total number of original keywords in those papers is 3,937, and average 4.34 keywords per paper. The distribution of those papers with different number of keywords is shown in Table 3. Generally, the number of keywords in each paper is about 3-6, covering 95.89% of this collection.

[2] Academic Papers: https://drive.google.com/open?id=1d2MHjGYPSDijgqeTBZQEPQ7EhB2ksHWl

Table 3. The distribution of academic papers with different number of keywords.

Number of keywords	Number of papers	Number of keywords	Number of papers
1	3	6	120
2	12	7	24
3	329	8	27
4	639	9	2
5	498	–	–

Dataset 3: A large amount of original Chinese patent dataset. The original Chinese patents are collected from SIPO, and the publication times of these patents range from 2016.11.01 to 2016.11.30. The number of well-structured Chinese patents is about 1.21 million, which is accessible in our google drive[3]. This patent dataset is used to,

- Train a word2vec model for Chinese patents.
- Generate an IDF dictionary for Chinese patents.
- As the source of manually annotated Chinese patent corpus.

Dataset 4: A human-annotated patent dataset, which consists of 839 Chinese patents. This dataset is manually annotated by three referees from the disciplinary of computer science with the following requirements.

- Assign 3-6 keywords for each Chinese patent.
- Keywords with 2-7 Chinese characters in length.
- Try to select the word whose POS is noun, verb or adjective.

At last, the union of pairwise intersections between the annotations are adopted as the human-annotated gold standard dataset for Chinese patents [22]. The human-annotated patent example is shown in Table 4.

From the results of human-annotated keywords, it is obvious that the annotated keywords are generally long phrases with specific meanings. However, this study primarily focuses on the keywords that make up these key phrases. In order to evaluate the performance of keywords extraction algorithm in a more critical way, two agreements on the keywords extraction are defined as follows:

- Exact Match: when two phrases match exactly.
- Relaxed Match: when two phrases either match exactly or can be made identical by adding a single word to the beginning or end of the shorter phrase [22].

Micro-averaged precision, recall and F-score under these two settings are calculated by the same formula as [18].

4.2 Parameters Selection

There are totally six parameters in SR based TF-IDF algorithm. Beside the damping factor d which is generally taken as 0.85 according to PageRank, there are five parameters left, including ε, α, β, K_S and K_W. α is used to adjust the weight between different heuristic rules, which are all treated equally in our algorithm, so that the α is taken as 0.5

[3] Original CN-Patents: https://drive.google.com/open?id=12nvcUbQ8XSXSBWzN9RXmBvHEjhJc6JQC

Table 4. The example of human-annotated Chinese patents.

Title	一种大数据环境下用户异常行为检测分析方法 Detection and analysis method for abnormal user behaviors in big data environment
Abstract	本发明涉及一种大数据环境下用户异常行为检测分析方法。其特点是，包括如下步骤：根据 HDFS 中历史一个统计周期内用户的日志记录，用户异常行为检测系统利用机器学习通过离线方式对用户访问行为进行异常分析，建立用户行为模型；基于 Storm 中当前实时用户操作行为，用户异常行为检测系统在线比较实时行为和历史行为的差异；若上述二者差异较大，则向 Kaffka 发送安全预警信息并在 Stream 界面中展示，否则，判断该行为是合规的安全行为。与现有技术相比，本发明通过机器学习算法支持根据用户在 Hadoop 平台上历史使用行为习惯来定义行为模式或用户画像的能力。默认该训练系统以每月的频率更新模型，模型粒度为一分钟。 The invention relates to user's Anomaly behaviors detecting analytical approach under the big data environment. Its characteristics are, including following step: user's log recording in the historical one statistics cycle in according to HDFS, user's Anomaly behaviors detecting system utilizes the machine learning to carry out the abnormal analysis by the off-line mode to the user access behaviors, sets up the personal behavior model; Based on current active user operation behavior among the Storm, the difference of online more real-time action of user's Anomaly behaviors detecting system and historical action; If above-mentioned the two differ greatly, then send safety precaution information and show in the Stream interface to Kaffka, otherwise, judge that the action is the safety behavior that closes rule. Compare with the prior art, the invention defines the ability of behavior pattern or user's portrait according to user's historical use behavioral habits on the Hadoop platform through machine learning algorithm prop root. Give tacit consent to the Training system with the frequency updating model of per mensem, the model granularity is one minute.
Annotated keywords	大数据；异常行为；机器学习；行为模型；用户画像；用户行为 Big data; abnormal behaviors; machine learning; behavior model; user's portrait; user's behaviors

for each rule. The remaining four parameters, ε, β, K_S and K_W, are discussed one by one in the following parts.

As seen in Algorithm 1, the parameter ε determines the convergence rate of SR based TF-IDF algorithm. The average number of iterations in SR based TF-IDF algorithm is illustrated in Fig. 4 when ε ranges in $[10^{-7}, 10^{-1}]$. Obviously, the algorithm submits excellent performance in numerical convergence, the number of iterations is still within 20 even when $\varepsilon = 10^{-7}$, so the value of ε is taken as 10^{-7}.

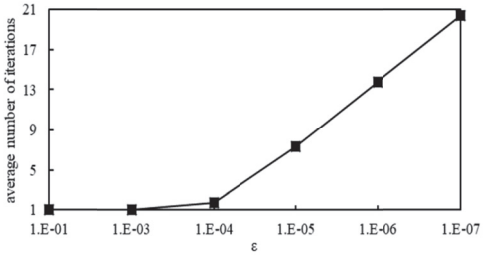

Fig. 4. The effect of parameter ε on convergence rate of SR based TF-IDF.

In order to discuss the values of remaining parameters (β, K_S and K_W), SR based TF-IDF algorithm is used to extract keywords from Chinese patents (**Dataset 4**), and the

F-score of exact-match and relaxed-match is calculated respectively with different candidate values of each parameter. The results are presented in Tables 5 and 6.

Table 5. F-score of relaxed-match.

Top-K_S (%)	Top-K_W	β					
		0.1	0.2	**0.3**	0.4	0.5	0.6
0.70	Top3	0.499	0.509	0.510	0.504	0.497	0.497
	Top4	**0.515**	**0.524**	**0.524**	**0.517**	**0.509**	**0.505**
	Top5	0.514	0.518	0.517	0.512	0.505	0.497
	Top6	0.507	0.505	0.502	0.497	0.490	0.483
0.75	Top3	0.494	0.505	0.512	0.505	0.498	0.497
	Top4	0.515	**0.522**	**0.525**	**0.521**	**0.510**	**0.505**
	Top5	**0.516**	0.520	0.517	0.513	0.504	0.497
	Top6	0.507	0.507	0.504	0.499	0.492	0.484
0.80	Top3	0.488	0.505	0.513	0.504	0.501	0.499
	Top4	0.513	**0.524**	**0.527**	**0.522**	**0.513**	**0.507**
	Top5	**0.517**	0.523	0.518	0.513	0.506	0.498
	Top6	0.505	0.508	0.507	0.499	0.493	0.484
0.85	Top3	0.486	0.503	0.512	0.504	0.502	0.498
	Top4	0.509	0.523	**0.529**	**0.524**	**0.514**	**0.507**
	Top5	**0.514**	**0.526**	0.520	0.513	0.506	0.499
	Top6	0.501	0.511	0.509	0.502	0.495	0.486
0.90	Top3	0.481	0.501	0.513	0.503	0.501	0.498
	Top4	0.503	0.520	**0.528**	**0.524**	**0.515**	**0.508**
	Top5	**0.509**	**0.524**	0.522	0.515	0.506	0.500
	Top6	0.499	0.510	0.510	0.503	0.496	0.487

Table 6. F-score of exact-match.

Top-K_S (%)	Top-K_W	β					
		0.1	0.2	**0.3**	0.4	0.5	0.6
0.70	Top3	0.204	0.217	0.212	0.210	**0.207**	**0.207**
	Top4	0.210	**0.219**	**0.217**	**0.212**	**0.207**	0.204
	Top5	**0.213**	0.213	0.211	0.207	0.201	0.197
	Top6	0.210	0.208	0.205	0.200	0.197	0.191
0.75	Top3	0.200	0.213	0.212	0.212	**0.209**	**0.207**
	Top4	0.209	**0.220**	**0.218**	**0.214**	0.208	0.204
	Top5	**0.213**	0.215	0.209	0.207	0.202	0.197
	Top6	0.209	0.209	0.206	0.201	0.198	0.192
0.80	Top3	0.199	0.213	0.214	0.213	0.210	**0.209**
	Top4	0.211	**0.221**	**0.218**	**0.216**	**0.211**	0.207
	Top5	**0.214**	0.217	0.210	0.208	0.204	0.199
	Top6	0.209	0.209	0.209	0.201	0.198	0.193
0.85	Top3	0.197	0.211	0.214	0.214	0.211	**0.210**
	Top4	0.208	**0.219**	**0.220**	**0.217**	**0.212**	0.207
	Top5	**0.213**	0.218	0.212	0.210	0.205	0.199
	Top6	0.208	0.213	0.209	0.203	0.199	0.194
0.90	Top3	0.192	0.210	0.215	0.215	0.211	**0.210**
	Top4	0.206	0.217	**0.221**	**0.218**	**0.212**	0.208
	Top5	**0.212**	**0.220**	0.213	0.209	0.205	0.200
	Top6	0.208	0.212	0.209	0.204	0.200	0.194

The relation between K_W and the optimal F-score is illustrated in Table 7, which is derived from Table 5 and Table 6. No matter what the values of β and K_S are, the F-score always gets much more optimal values when K_W is 4 (underlined-bold numbers in Tables 5 and 6). The conclusion of the analysis is strengthened by the average 4.34 keywords for each academic paper in **Dataset 2**. Therefore, K_W is taken as 4 for keywords extraction from Chinese patents.

Table 7. The relation between K_W and the optimal F-score.

K_W	3	**4**	5	6
Number of optimal F-score	7	**42**	12	0
Proportion of optimal F-score	0.115	**0.689**	0.197	0

With the fixed value $K_W = 4$ and random values of K_S, the F-score of relaxed-match achieves the most number of optimal values only when $\beta = 0.3$ (boxed-underlined-bold numbers in Table 5). However, the algorithm gets the best F-score on the metric of exact-match with the value of β is 0.2. To ensure the maximum coverage rate and the minimum average error of the optimal F-score both on the metrics of relaxed-match and exact-match (see Table 8), the final value of parameter β is 0.3. Similarly, the F-score on the metric of relaxed-match and exact-match achieves the global optimal value when $K_S = 0.85$. In addition, the value of K_S also indicates that the expression of patents is rigorous with less noise. In a word, by selecting the top-K_S percent sentences of the patents can enhance the performance of keywords extraction. The final value of K_S is 0.85 for keywords extraction from Chinese patents.

Table 8. The relation between β and the optimal F-score.

β	0.2	**0.3**
Coverage rate of optimal F-score	0.300	**0.700**
Average error of optimal F-score	0.0022	**0.0007**

Although the parameters are on the basis of the aforementioned discussion, this paper attempts to find a method that can automatically determine the values of these hyper-parameters. A succinct and effective method based on Grid Search [47] is proposed to discover the optimal parameters, and the pseudo-code is shown in **Algorithm 2**. In this algorithm, there are three parameters required to be selected (K_S, K_W, β). *Comb*(K_S, K_W, β), the combination of these three parameters, is $\{\{K_S\}, \{K_W\}, \{\beta\}, \{K_S, K_W\}, \{K_W, \beta\}, \{K_S, \beta\}, \{K_S, K_W, \beta\}\}$. *MSE* is short for Mean Square Error. By this means, the optimal values of K_S, K_W and β are also determined, which are same with the aforementioned discussion.

Algorithm 2: Grid Search

Input: K_S is the percent of patent sentences, its range is [0, 1] and the step is 0.05. K_W is the top number of keywords, its range is [3, 6] and the step is 1. β is the weight of heuristic rules mentioned in Section 3.3.2, its range is [0, 1] and the step is 0.1.

Output: The optimal values of O_K_S, O_K_W and O_β, which are initial with ϕ.

Initialize list *FScoreExactAll*, *FScoreRelatedAll*
for $K_S \leftarrow 0$ to 1:
  for $K_W \leftarrow 3$ to 6:
    for $\beta \leftarrow 0$ to 1:
      fscoreExact(K_S, K_W, β) ← F1-score of exact-match (**Algorithm 1)**;
      fscoreRelated(K_S, K_W, β) ← F1-score of related-match (**Algorithm1)**;
      FScoreExactAll append *fscoreExact*(K_S, K_W, β)
      FScoreRelatedAll append *fscoreRelated*(K_S, K_W, β)
      $\beta \leftarrow \beta + 0.1$;
    end for
  $K_W \leftarrow K_W + 1$;
  end for
$K_S \leftarrow K_S + 0.05$
end for
MSE ← 100
for *fscoreExact*(K_S, K_W, β) in *FScoreExactAll*:
  fscoreExactMax(*Comb*(K_S, K_W, β)) ← max value of *fscoreExact*(*Comb*(K_S, K_W, β)) in *FScoreExactAll*;
  fscoreRelatedMax(*Comb*(K_S, K_W, β)) ← max value of *fscoreRlated*(*Comb* (K_S, K_W, β)) in *FScoreRelatedAll*;
  $temp \leftarrow \sum_{comb \in Comb} (fscoreExact(K_S, K_W, \beta) - fscoreExactMax(comb))^2 + \sum_{comb \in Comb} (fscore\,\mathrm{Re}lated(K_S, K_W, \beta) - fscoreRelatedMax(comb))^2$
  if *temp* < *MSE*:
    MSE ← *temp*
    (O_K_S, O_K_W, O_β) ← (K_S, K_W, β)
  end if
end for

4.3 Results Comparison and Discussion

As discussed in Section 4.2, the parameters of SR based TF-IDF algorithm adopt the following values: $d = 0.85$, $\varepsilon = 10^{-7}$, $\alpha = 0.5$, $\beta = 0.3$, $K_S = 0.85$ and $K_W = 4$. In order to verify the effectiveness and superiority of our approach, SR based TF-IDF algorithm is evaluated with other four keywords extraction algorithms, including the commonly used TF-IDF, TextRank and the latest word2vec weighted TextRank (2017), PKEA (2018). For a fair comparison between the five keywords extraction algorithm, we conduct each experiment under the same conditions. Table 9 shows the experimental results achieved by these five algorithms on precision, recall and F-score under the metrics of exact-match and relaxed-match. For better understanding the effectiveness of the proposed approach, some extracted keywords examples of these five algorithms are listed in Table 10. Owing to limited space, the titles and abstracts of the example patents are not listed here, but you can find the detailed information in our public **Datasets 4** by the patent ids.

As shown in Table 9, the results of the commonly used TF-IDF algorithm and TextRank algorithm are well match on the metrics of exact-match (13.9% and 15.0% in F-score) and relaxed-match (48.4% and 48.5% in F-score). The TF-IDF algorithm is simple but effective, and the result is in line with the existing researches. However, because of the rigorous usage of language and text expression in patent texts, there are a lot of candidate keywords with *low frequency* (For examples, in the patent CN201410062329.0 (Table 10), the keyword "机器学习" (machine learning) only appears one time with the non-keyword "样本" (samples) appears 9 times.) or *noise words* ("所述" (said) in patent CN201610815864.8), which cannot be distinguished by the TF-IDF algorithm.

Table 9. The performance of keywords extraction algorithm from Chinese patents.

Method	exact-match			relaxed-match		
	P	R	F	P	R	F
TF-IDF	0.147	0.134	0.139	0.514	0.467	0.484
TextRank	0.159	0.145	0.150	0.515	0.468	0.485
word2vec weighted TextRank (2017)	0.168	**0.252**	0.202	0.487	0.505	0.496
PKEA (2018)	0.151	0.134	0.141	0.329	0.295	0.308
SR based TF-IDF	**0.235**	0.213	**0.220**	**0.562**	**0.511**	**0.529**

In TextRank, the problem has a little improved (a 1.1% rise of exact-match and a 0.1% rise of relaxed-match in F-score) due to the consideration of co-occurrence relations between words in TextRank graph. Nevertheless, the low co-occurrence of words in abstracts of Chinese patents leads to a sparse words graph, which cannot make good use of the connectivity of network to transmit the weights between words. In order to improve the sparsity of the words graph, a word2vec weighted TextRank [18] is proposed to enrich the semantic relations between words, which tremendously leverage the precision, recall and F-score both on exact-match and relaxed-match. Especially the recall on the metric of exact-match is up to 25.2% and rises over 10% than TF-IDF and TextRank. It firmly supports that word2vec can represent the potential semantic information for Chinese patents, and also proves the effectiveness of semantic information for patent keywords extraction. All of these discoveries are considered in our SR based TF-IDF algorithm.

The other latest keywords extraction algorithm compared in this paper is PKEA [25]. However, for the task of Chinese patent keywords extraction, the PKEA algorithm bears unsatisfactory performance compared with our SR based TF-IDF algorithm (30.8% vs. 52.9% in F-score on relaxed-match), even gets worse than the commonly used TF-IDF algorithm (30.8% vs. 48.4% in F-score on relaxed-match). This is because the PKEA algorithm aims to extract effective features for patent classification. To a large extent, the extracted keywords are similar with the centroid word of each patent category. The samples of patent keywords listed in Table 10 appears to confirm it, the keywords of these examples are the category keywords of machine leaning (CN201410062329.0), public transportation (CN201610935591.0) and cloud computing (CN201610815864.8). In other words, the keywords extracted by PKEA are the category keywords of each patent, not the content keywords.

Table 10. Samples of Keywords Extraction from Chinese patents.

Patent ID	CN201410062329.0	CN201610935591.0	CN201610815864.8
Human-annotated	机器学习/machine learning 高光谱/hyperspectral 地物分类/ground targets classification	智能手机/smartphone 公共汽车/bus 舒适性/comfort 乘车/riding 指标/index	云计算/cloud computing 树结构/tree structure 操作方法/operational approach 节点序列/node sequence
TF-IDF	样本/sample 地物/ground targets 分类/classification 计算/compute	信息/information 舒适性/comfort 公共汽车/bus 车辆/vehicle	节点/node 序列/sequence 树结构/tree structure 所述/said
TextRank	样本/sample 地物/ground targets 标记/label 高光谱/hyperspectral	舒适性/comfort 车辆/vehicle 运行/move 乘车/riding	节点/node 序列/sequence 树结构/tree structure 标识符/identifier
word2vec weighted TextRank (2017)	机器学习/machine learning 聚类/clustering 高光谱/hyperspectral 修正/correction	降维/dimensionality reduction 指标/index 乘车/riding 运行状态/moving status	云计算/cloud computing 系统/system 关系表/relational table 树结构/tree structure
PKEA (2018)	机器学习/machine learning 模型预测/model prediction 隐藏层/hidden layer 聚类/clustering	信息/information 经纬度/ latitude-longitude 车辆/vehicle 站点/station	云计算/cloud computing 信息/information 查询/query 树结构/tree structure
SR based TF-IDF	高光谱/hyperspectral 地物/ground targets 分类/classification 机器学习/machine learning	公共汽车/bus 舒适性/comfort 智能手机/smartphone 乘车/riding	操作方法/operational approach 树结构/tree structure 节点/node 云计算/cloud computing

In a word, it is effective for keywords extraction to select top-K_S percent sentences and enrich the semantic relationship between words. The SR based TF-IDF algorithm proposed in this work selects the candidate sentences by a sentence-ranking model and transmits the semantic weights of sentences into the candidate words. Therefore, the F-score on the metrics of exact-match and relaxed-match based on the SR based TF-IDF algorithm achieves the highest respectively (a maximum 8.1 % rise of exact-match and a maximum 22.1% rise of relaxed-match in F-score). However, the human-annotated patent keywords are always the special significant key-phrases, such as "地物分类" (ground targets classification) in patent CN201410062329.0 from Table 10. Our algorithm mainly focuses on the keywords that make up those key-phrases, which causes certain limitations of extraction results (For examples, the keywords "地物分类" (ground targets classification) appears as "地物" (ground targets) and "分类" (classification) in the results of our algorithm). Whereas this problem will be taken into consideration in our future work.

4.4 Additional Evaluations

In this section, an additional experiment is adopted to evaluate the proposed algorithm on public Chinese and English datasets respectively. Section 4.4.1 evaluates the five keywords extraction algorithms on a Chinese public dataset of academic papers (**Dataset 2**) to illustrate the applicability of our approach to other fields. Moreover, the experiments on *Inspec* [14] (**Dataset 1**) shows that the proposed approach can scale and generalize well to English language beyond the Chinese language (See Section 4.4.2).

4.4.1 Experiments on Chinese academic papers

In order to check up the field applicability, our proposed algorithm is used to extract keywords from the academic papers beyond patents. Moreover, the parameters of our proposed algorithm in this experiment are selected like the Section 4.2. The parameters of SR based TF-IDF algorithm in this dataset are: $d = 0.85$, $\varepsilon = 10^{-7}$, $\alpha = 0.5$, $\beta = 0.2$, $K_S = 0.90$ and $K_W = 4$. Table 11 shows the experimental results achieved by the above-mentioned five algorithms on precision, recall and F-score under the metrics of exact-match and relaxed-match.

Table 11. Performance comparison on Chinese academic papers.

Method	exact-match			relaxed-match		
	P	R	F	P	R	F
TF-IDF	0.077	0.074	0.075	0.412	0.392	0.395
TextRank	0.071	0.085	0.077	0.383	**0.453**	0.409
word2vec weighted TextRank (2017)	0.107	**0.122**	0.112	0.428	0.406	0.410
PKEA (2018)	**0.120**	0.110	0.113	0.292	0.271	0.278
SR based TF-IDF	0.119	0.113	**0.114**	**0.451**	0.429	**0.433**

In the experiments on Chinese academic papers (Table 11), the best result of recall on exact-match and relaxed-match is achieved by TextRank and word2vec weighted TextRank respectively, and the precision on exact-match is achieved by PKEA. However, there are only a little bit gap between our algorithm and the other keywords extraction algorithms. Besides, the optimal F-scores are achieved by our SR based TF-IDF algorithm, around 11.4% on exact-match and 43.3% on relaxed-match, which outperforms the other algorithms for at least 2.3%. The results on Chinese academic papers further validate the thought of "the keywords are in the key sentences", and also validate that our proposed algorithm has generalization capability to extract keywords from the other types of texts.

4.4.2 Experiments on *Inspec* [14]

Paper [14] provides *Inspec* dataset and evaluation methods. We apply our SR based TF-IDF and the other four keywords extraction algorithms (TF-IDF, TextRank, word2vec weighted TextRank and PKEA) to this dataset (see **Dataset 1**) and list the experimental results in Table 12. Moreover, the parameters of SR based TF-IDF algorithm are:

$d = 0.85$, $\varepsilon = 10^{-7}$, $\alpha = 0.5$, $\beta = 0.1$, $K_S = 0.90$ and $K_W = 11$. In the experiments on this dataset, the F-score achieved by the commonly used TF-IDF algorithm is around 4.8% on exact-match, which outperforms our method for 2%. However, under the metric of relaxed-match, our algorithm outperforms the other four methods for at least 3.2%, even has a 10% rise than TF-IDF algorithm. The experiment further verifies that our algorithm concentrates on the keywords more than key-phrases. Nevertheless, the experimental results on the *Inspec* [14] dataset shows that our SR based TF-IDF algorithm is capable of extracting keyword from texts in not only Chinese but also other languages.

Table 12. Performance comparison on *Inspec* [14].

Method	exact-match			relaxed-match		
	P	R	F	P	R	F
TF-IDF	**0.044**	**0.061**	**0.048**	0.302	0.440	0.337
TextRank	0.025	0.040	0.029	0.366	0.506	0.400
word2vec weighted TextRank (2017)	0.021	0.031	0.023	0.312	0.444	0.346
PKEA (2018)	0.020	0.025	0.022	0.304	0.289	0.296
SR based TF-IDF	0.025	0.035	0.028	**0.420**	**0.510**	**0.432**

5. CONCLUSIONS

Keywords extraction from Chinese patents is a widely open research issue, with huge potential benefits given the continual growing number of Chinese patents. This paper presents a new method for extracting keywords from Chinese patents, inspired by the thought of "the keywords are in the key sentences". Our method learns a sentence-ranking model based on a sentence semantic graph and heuristic rules to filter top-K_S percent sentences for each patent. The experimental results on real Chinese patents show that SR based TF-IDF algorithm outperforms four baseline keywords extraction algorithms, including the commonly used TF-IDF, TextRank and the latest word2vec weighted TextRank (2017), PKEA (2018). In addition, two more experiments on public Chinese and English datasets further testify that our SR based TF-IDF algorithm has a generalization capability in keywords extraction for texts both in other type and languages.

For further improvement, more features, including the dependency grammar, will be supplemented to train sentence embedding. Moreover, the theory of information entropy will also be introduced into the SR based TF-IDF algorithm to merge the keywords and generate more proper and practical key-phrases from the Chinese patents.

REFERENCES

1. C. Liu and C. Xu, "The application of patent mining in the forecast of smart home industry," *Lancet*, Vol. 10, 2016, pp. 67-75.
2. Patent Search and Analysis of SIPO, "Data inclusion range," http://www.pss-system.gov.cn/sipopublicsearch/portal/uiInitPortalHome-showDataRange.shtml, 2017.
3. SIPO, "The summary statistics of the world's top five IPO in 2016," http://www.sipo.gov.cn/docs/pub/old/tjxx/wjndbg/201704/P020170425316456439271.pdf, 2017.

4. C. Wu, "Constructing a weighted keyword-based patent network approach to identify technological trends and evolution in a field of green energy: a case of biofuels," *Quality & Quantity*, Vol. 50, 2016, pp. 213-235.
5. J. Joung and K. Kim, "Monitoring emerging technologies for technology planning using technical keyword based analysis from patent data," *Technological Forecasting and Social Change*, Vol. 114, 2017, pp. 281-292.
6. E. Uzun, H. V. Agun, and T. Yerlikaya, "A hybrid approach for extracting informative content from web pages," *Information Processing & Management*, Vol. 49, 2013, pp. 928-944.
7. N. Li, J. Wang, H. Bai, *et al*, "Research and implementation of FFT-based extraction algorithm of webpage," *Computer Applications in Engineering Education*, Vol. 43, 2007, pp. 148-151.
8. H. Zhang, "Chinese key words extraction algorithm," *Computer Systems & Applications*, Vol. 18, 2009, pp. 73-76.
9. P. Qin, H. Zhang, and J. Liu, "Keyword extraction based on new word detection," *Control & Automation*, Vol. 26, 2010, pp. 257-258.
10. I. H. Witten, G. W. Paynter, E. Frank, *et al.* "KEA: Practical automatic keyphrase extraction," in *Proceedings of ACM Conference on Digital Libraries*, 1998, pp. 254-255.
11. A. Csomai and R. Mihalcea, "Investigations in unsupervised back-of-the-book indexing," in *Proceedings of the 20th International Florida Artificial Intelligence Research Society Conference*, Vol. 16, 2007, pp. 231-242.
12. Y. Wei, D. Fontaine, and J. P. Barthes, "Automatic keyphrase extraction with a refined candidate set," in *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Vol. 1, 2009, pp. 576-579.
13. Z. Liu, P. Li, Y. Zheng, *et al.*, "Clustering to find exemplar terms for keyphrase extraction," in *Proceedings of Conference on Empirical Methods in Natural Language Processing*, Vol. 1, 2009, pp. 257-266.
14. A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in *Proceedings of Conference on Empirical Methods in NLPACL*, 2003, pp. 216-223.
15. R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," in *Proceedings of Conference on Empirical Methods in Natural Language Processing*, 2004, pp. 404-411.
16. Z. Liu, W. Huang, Y. Zheng, *et al.* "Automatic keyphrase extraction via topic decomposition," in *Proceedings of Conference on Empirical Methods in NLPACL*, 2010, pp. 366-376.
17. W. Li and J. Zhao, "TextRank algorithm by exploiting Wikipedia for short text keywords extraction," in *Proceeeings of International Conference on Information Science and Control Engineering*, 2016, pp. 683-686.
18. Y. Wen, H. Yuan, and P. Zhang, "Research on keyword extraction based on word2-vec weighted TextRank," in *Proceedings of International Conference on Computer and Communications*, 2017, pp. 2109-2113.
19. S. K. Biswas, M. Bordoloi, and J. Shreya, "A graph based keyword extraction model using collective node weight," *Expert Systems with Applications*, Vol. 97, 2018, pp. 51-59.
20. K. S. Hasan and V. Ng, "Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art," in *Proceedings of International Conference on Com-*

putational Linguistics, Posters, 2010, pp. 365-373.
21. Q. Zhang, D. Xue, Z. Zhang, *et al.*, "Automatic keyword extraction from massive data sets based on feature combination," *Journal of the China Society for Scientific and Technical Information*, Vol. 25, 2006, pp. 587-593.
22. S. Lahiri, R. Mihalcea, P. H. Lai, "Keyword extraction from emails," *Natural Language Engineering*, Vol. 23, 2017, pp. 295-317.
23. O. Medelyan, E. Frank, and I. H. Witten, "Human-competitive tagging using automatic keyphrase extraction," in *Proceedings of Conference on Empirical Methods in Natural Language Processing*, 2009, pp. 1318-1327.
24. H. Chen, G. Zhang, D. Zhu, *et al.*, "Topic-based technological forecasting based on patent data: A case study of Australian patents from 2000 to 2014," *Technological Forecasting & Social Change*, Vol. 119, 2017, pp. 39-52.
25. J. Hu, S. Li, Y. Yao, *et al.*, "Patent keyword extraction algorithm based on distributed representation for patent classification," *Entropy*, Vol. 20, 2018, p. 104.
26. M. Cao, R. Chen, J. Sun, *et al.*, "Comparative research on technology competitiveness based on patent analysis," *Studies in Science of Science*, Vol. 34, 2016, pp. 380-385.
27. W. Ding, Y. Liu, and J. Zhang, "Chinese-keyword fuzzy search and extraction over encrypted patent documents," in *Proceedings of International Joint Conference on Knowledge Discovery*, *Knowledge Engineering and Knowledge Management*, Vol. 1, 2015, pp. 168-176.
28. Y. Chen, R. Zhou, W. Zhu, *et al.*, "Mining patent knowledge for automatic keyword extraction," *Journal of Computer Research and Development*, Vol. 53, 2016, pp. 1740-1752.
29. D. Liu, Z. Peng, B. Liu, *et al.*, "Technology effect phrase extraction in Chinese patent abstracts," in *Proceedings of Asia-Pacific Web Conference*, Vol. 8709, 2014, pp. 141-152.
30. D. Liu and L. Wang, "Keywords extraction algorithm based on semantic dictionary and lexical chain," *Journal of Zhejiang University of Technology*, Vol. 41, 2013, pp. 545-551.
31. Sogou.com, "The thesaurus of Sogou input method," https://pinyin.sogou.com/d-ict/cate/index/96?rf=dictindex, 2018.
32. Baidu, "Science Baike," https://baike.baidu.com/science, 2018.
33. HIT-SCIR, "LTP," https://www.ltp-cloud.com/, 2018
34. J. Yu and Y. Dang, "Chinese term extraction based on POS analysis & string frequency," *Systems Engineering-Theory & Practice*, Vol. 30, 2010, pp. 105-111.
35. W. Wang, Z. Li, J. Wang, *et al.*, "How far we can go with extractive text summarization? Heuristic methods to obtain near upper bounds," *Expert Systems with Applications*, Vol. 90, 2017, pp. 439-463.
36. B. Sharifi, M. A. Hutton, and J. Kalita. "Summarizing microblogs automatically," in *Proceedings of Annual Conference of the North American Chapter of the ACL*, 2010, pp. 685-688.
37. L. Zhang, J. Cao, C. Pu, *et al.*, "Single document automatic summarization algorithm based on word-sentence co-ranking," *Journal of Computer Applications*, Vol. 37, 2017, pp. 2100-2105.
38. Z. Lin, M. Feng, C. N. D. Santos, *et al.*, "A structured self-attentive sentence embed-

ding," arXiv preprint arXiv:1703.03130, 2017.
39. H. Palngi, L. Deng, Y. Shen, *et al.*, "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval," *Transactions on Audio, Speech, and Language Processing*, Vol. 24, 2016, pp. 694-707.
40. M. Tan, C. D. Santos, B. Xiang, *et al.*, "Improved representation learning for question answer matching," *Meeting of the Association for Computational Linguistics*, 2016, pp. 464-473.
41. Google, "Word2vec, " https://code.google.com/archive/p/word2vec/, 2013.
42. X. Rong, "Word2vec parameter learning explained," arXiv preprint arXiv:1411.2738, 2014.
43. J. Zhang, "Method for the extraction of Chinese text keywords based on multi-feature fusion," *Journal of Information Studies:Theory and Application*, Vol. 10, 2013, pp. 105-108.
44. Z. Zhang, "Research on the modular Chinese sentence similarity computing based on hownet," Department of Computer Science and Technology, Anhui University of Technology, 2010.
45. P. B. Baxendale, "Machine-made index for technical literature: an experiment," *IBM Corp.*, Vol. 2, 1958, pp. 354-361.
46. Wanfang Data Co., Ltd., "Wanfang data e-resources for China studies," http://www.wanfangdata.com.cn/index.html, 2018.
47. R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of International Joint Conference on Artificial Intelligence*, Vol. 14, 1995, pp. 1137-1143.

Zhi-Hong Wang (王志宏) was born in Jiangsu, China in 1990. He received the M.S. degrees in Computer Science and Engineering from East China University of Science and Technology, Shanghai, China. He is currently pursuing a Ph.D. degree in Computer Science and Technology from East China University of Science and Technology. His research interests include text mining, machine learning and big data analysis.

Yi Guo (过弋) received this MS degree in Computer Science from Xidian University, Xi'an, China and Ph.D. degree in Computer Science from Heriot-Watt University, Edinburgh, Scotland in 2005. He is currently a Professor at East China University of Science and Technology. His research focus is text mining, information extraction, knowledge discovery and business intelligence analysis. Now he is the member of MIEEE MCMI MIET MBCS and APMG-MSP/PRINCE2-Practitioner, and he is also a committee member of National Engineering Laboratory for Big Data Distribution and Exchange Technologies.