Keywords Extraction based on Sentence-Ranking from Chinese Patents

Zhihong Wang*, Yi Guo*†‡, Tianmei Qi*
*Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai, China
†School of Information Science and Technology, Shihezi University, Xinjiang, China

***Abstract*—Patent, an important scientific literature, records a large amount of innovative and practical research. The patent keywords also provide a high-level topic description of a patent document and hold an important position in classic NLP tasks, such as patent classification or clustering. However, there are few research works on keywords extraction covering the Chinese patents in current stage. In this paper, we propose a novel algorithm to extract keywords from Chinese patents. A sentence-ranking model, based on a sentence embedding graph and heuristic rules, is constructed to select the top-K_S percent of the sentences. At the same time, the semantic-ranking weights of sentences are also transmitted to keywords extraction. The experimental results on our Chinese patents datasets testifies that the sentence-ranking based keywords extraction algorithm improves the performance by 6% to 13% in F-score. In summary, the new idea of selecting key sentences from original documents can effectively filter out noisy sentences and leverage the performance of keywords extraction.**

***Index Terms*—Chinese patents, key sentences, sentence-ranking, keywords extraction**

I. Introduction

Chinese patent documents has reached 40,673,532 till October 2017 according to the latest statistics of State Intellectual Property Office (SIPO), near a third of global patent documents. Meanwhile, it keeps high growth rates every year, such as the growth rate reached 9% in 2017. Patent is a kind of important scientific literature, which records a large amount of innovative and practical research productions in industry and academia. And we can also speculate on the direction of new technologies and even develop new application areas by analyzing patent bibliographic, changes in patent legal status or citation relations [1]. All in all, it is of great importance to analyze and mine valuable information from mass Chinese patent documents.

Patent keywords provide a high-level topic description of a patent document and play an important role in many applications or tasks of patents. For example, Fujii [2] stated that keywords play a key factors' role in patent translation. Thus, patent keywords get more and more attentions and are widely applied. However, all Chinese patent documents have no author-assigned keywords, which make manually assigning keywords for each patent document very laborious jobs. Therefore, it is highly desirable to extract keywords automatically.

‡Corresponding author: guoyi@ecust.edu.cn
DOI reference number: 10.18293/SEKE2018-034

In this paper, we address the task of automatic keywords extraction from Chinese patents and propose an automatic keywords extraction system for this end and our contributions are as following:

- Construct a sentence-ranking model based on a sentence embedding graph and heuristic rules.
- Optimize the state-of-the-art keywords extraction system (TF-IDF) for Chinese patents based on the sentence-ranking model. The keywords extraction system is named SR based TF-IDF, short for Sentence-Ranking based Term Frequency and Inverse Document Frequency.

The rest of this paper is organized as follows: Section II describes the closely related work; Section III details the architecture of our keywords extraction system; Section IV evaluate our models with dedicated experiments and Section V concludes this paper.

II. Related Work

Quite a few research works have published about keywords extraction. In general, keywords extraction of Chinese text usually proceeds in four steps: pre-processing, candidate selection, keywords extraction, and post-processing.

In the first pre-processing step, the title and text will be extracted based on specified heuristics rules or extraction algorithms of content main body. At the same time, long texts should be segmented into several paragraphs with paragraph marks (carriage return character, line feeds etc.), and paragraphs sometimes need to be segmented into several sentences with punctuation [3]. In addition, Chinese does not have a clear demarcation between words like English. Some basic operations therefore are needed in the pre-processing step, such as word segmentation, part of speech tagging, new word detection [4] and so on.

The second phase is to determine the keywords candidate collection. So that the remaining steps of keywords extraction is no longer necessary to consider the features of non-candidate words, which will improve the efficiency of keywords extraction. In practice, there are some efficient optimization for candidate extraction. In the KEA algorithm, the candidate keywords are obtained by several basic rules such as the length of keywords, proper nouns and the characteristics of keywords. Csomai et al. [5] experimented with stop-word-filtered n-grams and named entities as potential keywords. However, candidate keywords are still with a wide range and

contain many non-grammatical phrases after selecting with the above rules. So that W. You and D. Fontaine et al. [6] proposed a method to reduce the range or noise of the candidate words. In this method, the top-k words with highest frequencies are defined as the core words, and then the associated words are added into the candidates by word co-occurrence with core words.

The third step - keywords extraction - is quite complicated, because it is not obvious to choose which extraction algorithms (ranking or classification). The most known keywords extraction algorithms are graph-ranking-based algorithms [7] which are derived from PageRank, such as TextRank. Based on the traditional TextRank algorithm, Nan et al. [8] proposed an eccentricity and degree centrality based complex network for keywords extraction, and Li et al. [9] used K-proximity coupled graph to transfer patents into complex graph model and a patent comprehensive correlation calculation method for quantitative analysis of keyword importance is proposed. [10] pointed out that the external knowledge base can be used to enrich the information to assist in keywords extraction for essay texts in TextRank algorithm.

Besides, the classification algorithms are also efficient in keywords extraction even better have better performance in some fields. In a study by Hasan and Ng [11], TF-IDF was shown to be a surprisingly robust candidate and beaten other more complex ranking strategies. Other important features for keywords classification include TF-IDF, first occurrence position of the word, word diameter, word length and is-in-title etc.

The final important step in keywords extraction is post-filtering, such as filtering short words, limiting the number of two Chinese words [12]. And adjacent words are also sometimes collapsed into phrases, for a more readable output.

Now, the research works on patents mainly focus on patent translation, patent retrieval and patent classification etc. And some other studies primarily concentrate on the research of technological competitiveness about enterprises, industries or regions [13]. By studying the patent through macro, middle or micro aspects, a multi-tridimensional comprehensive evaluation system of technological competitiveness is formed to compare the technological competitiveness among enterprises or other institutions. With the protection of intellectual property rights getting more and more attentions in China, domestic researchers have begun to study the Chinese patent documents. In a study of Liu [14], a semi-automatic patented-technical phrase extraction method was proposed, which achieved good results on Chinese patents and effectively reduced labor cost. In order to improve the patent retrieval speed, [15] linked all the patents by keywords, which were extracted based on an improved TF-IDF algorithms. Moreover, patents are also used to as a background knowledge base, which is helpful to design or realize a better keywords automatic extraction algorithm in other fields [16].

In summary, keywords extraction has made great achievements. However, we are aware that there are only few previous studies about keywords extraction algorithm for Chinese patents. Therefore, we introduce an novel keywords extraction framework for Chinese patents, and obtain the higher precious, recall and F-measure than the state-of-the-art algorithms or the latest keywords extraction algorithms.

III. Automatic Keywords Extraction from Chinese Patents

A. Overview of the Framework

The framework of keywords extraction from Chinese patents is shown in Fig. 1, which consists of a domain dictionary construction module and a keywords extraction module, and will be detailed next.

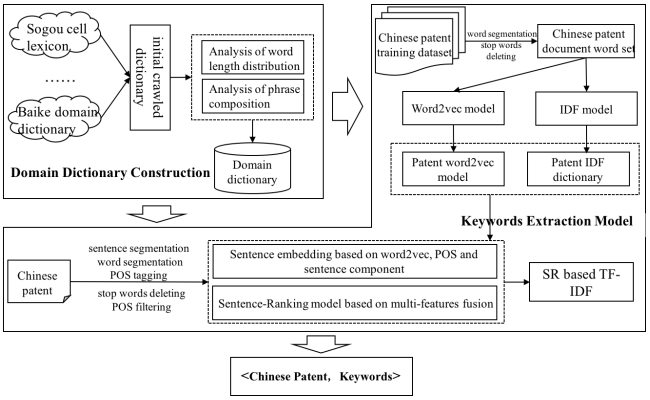

Fig. 1: The framework of keywords extraction from Chinese patents.

B. Domain Dictionary Cunstruction

Current keywords extraction algorithms for Chinese texts rely on word segmentation. The higher the quality of word segmentation is, the better results of keywords extraction we shall have. Thus, some new techniques based on external dictionaries [16] or new word detection [17] have been proven to improve the accuracy of Chinese word segmentation and contribute in keywords extraction. Chinese patents obviously contain a large number of professional terms. In order to elevate keywords extraction, a domain dictionary is constructed through merging multi-source heterogeneous external lexicons. 861 lexicons under the "Engineering Application" of Sogou cell lexicon and all vocabulary entries under the scientific classification from Baidu baike are collected, and we get 130 million words (or phrases) in total. After analyzing the initial lexicons, there are several problems emerging in the dictionary, such as plenty of duplicate vocabulary entries with the same meaning caused by English case or Chinese simplified and traditional, or a large number for combined-words.

Several means are used to tackle these above issues in this paper to achieve a higher quality domain dictionary. Firstly, all words in the dictionary are converted into normalized form, that is English in uppercase form and Chinese in simplified form, and delete the repeated entries. Then we analyze the distribution of word length of all words in the dictionary, and keep the words with length from 2 to 7, which is accounted for

0.906. Finally, a word segmentation tool, such as LTP, is used to segment into the most granular words and tag POS for the dictionary words. According to the Chinese word combination rules [18], 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 vocabulary entries, which contain these ban-words, will be deleted. Eventually 284,328 words are obtained after the above process.

C. Sentence-Ranking based Keywords Extraction Model

A Chinese patent keywords extraction model is proposed based on sentence-ranking model, which integrates the semantic graph and heuristic rules between sentences. The hidden idea is that the keywords is in the key sentences. That is to say, the bigger number of important words (such as keywords) a sentence has, the more important the sentence is. Correspondingly, the more frequently a word appears in an important sentence, the higher the importance of the word is, which is more likely to be the keyword of the document.

Generally, a sentence graph will be built to sort these sentences by the graph sorting algorithm, such as PageRank. In this paper, we introduce a sentence embedding model to better describe the semantic similarity between sentences, and several heuristic rules are also applied to sort sentences. Finally, the top-K_W keywords are extracted by improved the TF-IDF algorithm from the top-K_S percent candidate sentences with highest scores.

1) Sentence Embedding: Language models are a very important part of natural language processing, including the classic N-Gram model and the recently widely discussed deep learning model. Word2vec, a word embedding based on deep learning model, 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 assigned a corresponding vector in the space. To our knowledge, there is no good sentence embedding model until now. In order to vectorize a sentence, this paper attempts to propose a sentence embedding model on the basis of combining word embeddings produced by word2vec with sentence structural feature with primarily considering the factors, including sentence is a collection of words, words with different POS have different contributions to the sentences and words in different sentence composition have different contributions to the sentences.

For a sentence S, SW=(sw_1, sw_2, ..., sw_m), where sw_i(i=1, 2, ..., m) is the i-th word in sentence S. The corresponding POS of SW is denoted as SN=(sn_1, sn_2, ..., sn_m), where sn_i(i=1, 2, ..., m) represents the POS of the corresponding word sw_i(i=1, 2, ..., m). The word sw_i(i=1, 2, ..., m) in SW can be represented by a k*1 vector based on word2vec, name V_{sw_i}=$[v_{i1}, v_{i2}, ..., v_{ik}]^T$, where $v_{ij} \in \mathbf{R}$, j=1, 2, ..., k. Thus, the sentence embedding model of this paper 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 i-th word in sentence S and V_{sw_i} is its word embedding produced by word2vec. w_{pos} is the weight of word with different POS, and

$$w_{pos} = \begin{cases} 0.8, & if\ sn_i\ is\ noun. \\ 0.5, & if\ sn_i\ is\ verb. \\ 0.4, & if\ sn_i\ is\ adj. \\ 0, & if\ sn_i\ is\ others. \end{cases}$$

w_{sc} is the weight of word in different composition, and

$$w_{sc} = \begin{cases} 0.5, & if\ sw_i\ is\ subject. \\ 0.2, & if\ sw_i\ is\ predicate. \\ 0.3, & if\ sw_i\ is\ object. \\ 0, & if\ sw_i\ is\ others. \end{cases}$$

2) Multi-feature Fusion based Sentence-Ranking Model: This paper firstly gives the symbolic definitions of some variables and formally describes the problems of sentence-ranking in Chinese patents scenario. Suppose that for a Chinese patent document P, the title is T. The abstract sentences in P is S, and $|S| = n$. The goal of sentence-ranking model is to compute an n-dimensional vector SR=[SR_1, SR_2, ..., SR_n]^T, where SR_i is the weight of the i-th sentence. So that top-K_S percent sentences with the highest score can be obtained from SR.

(1) Heuristic Rules

Patents, a kind of scientific literature, have strict and standardized templates and writing criterions. The patent name presents the subject and type of the patent in a brief and accurate manner. While, the patent abstract clearly states the technical field to which the patent belongs, the technical issues to be solved by the patent, and the primary technical characteristics and uses of the patent. According to the analysis of patent, the following heuristics rules are considered:

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

Suppose that S_i is the i-th sentence in the set S of patent abstract sentences, and that SW_i = (swi_1, swi_2, ..., sw_{im}) represents all words in the sentence S_i, where sw_{ij} (j=1, 2 , ..., n) is the j-th word in the sentence S_i. All words in patent title T are represented by TW = (tw_1, tw_2, ..., tw_t), where tw_i (i=1, 2, ..., t) is the i-th word in title T.

Thus, the similarity between the patent title T and the sentences in patent abstract S is shown in formula 2.

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

where to_i is the similarity between the i-th sentence in patent abstract with the patent title T, which is calculated by Jaccard similarity,

Meanwhile, the weight of different position of patent abstract sentences is defined in formula 3.

$$W_{location}(S) = [loc_1, loc_2, \cdots, loc_n]^T \tag{3}$$

where loc_i is the location weight of the i-th abstract sentence in S. According to the sampling statistics of P.E.Baxendale, 85% of the sentences, which 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 are defined as follows.

when $n > 2$, define loc_i by

$$loc_i = \begin{cases} 0.85, & if\ S_i\ is\ the\ first\ sentence. \\ 0.07, & if\ S_i\ is\ the\ last\ sentence. \\ \frac{0.08}{n-2}, & if\ S_i\ isthe\ other\ sentences. \end{cases} \tag{4}$$

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

$$loc_i = \begin{cases} 0.89, & if\ S_i\ is\ the\ first\ sentence. \\ 0.11, & if\ S_i\ is\ the\ last\ sentence. \end{cases} \tag{5}$$

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

$$loc_i = 1 \tag{6}$$

According to formula 2 and 3, the weight of patent abstract sentences based on heuristic rules proposed in this paper is defined as 7.

$$loc_i = \alpha * W_{TitleOverlap}(S,T) + (1-\alpha) * W_{location}(S) \tag{7}$$

where, α is one of the weight parameter of the heuristic rules, the other is 1-α. While $W_{TitleOverlap}(S, T)$ is calculated by formula 2 and $W_{location}(S)$ is calculated by 3.

(2) A Sentence Semantic Graph

By considering the potential semantic information between sentences, a sentence semantic graph named G is built based on sentence embeddings, which takes the sentences in patent abstract as the vertex and the similar relation between the sentences as the edges. And PageRank is selected as the graph sorting algorithm in this paper. The adjacency matrix of semantic similarity between sentences is defined as $P_{sim}(S) = [S_{ij}]_{n*n}$, where S_{ij} is the weight of (S_i, S_j), which is defined by the semantic similarity between the i-th and the j-th sentence in patent abstract sentences S.

And the cosine similarity based on sentence embedding 8 is used to calculate the weight of edges in this paper.

$$s_{ij} = cos(V_{S_i}, V_{S_j}) = \frac{V_{S_i} \bullet V_{S_j}}{\|V_{S_i}\| \|V_{S_j}\|} \tag{8}$$

where S_i is the i-th sentence. V_{S_i} is the i-th sentence embedding, which is calculated by formula 1.

The iterative formula based on the idea of PageRank, which is used to achieve the sentence weight on the sentence semantic graph G, is as follows:

$$w_{PR}(S_i) = (1-d) + d * \sum_{s_{ij} \neq 1} [(\frac{s_{ij}}{\sum_k s_{ik}}) w_{PR}(S_j)] \tag{9}$$

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 i-th iteration, then the formula 9 can be re-expressed as:

$$W_{PR}^i = PW_{PR}^{i-1} \tag{10}$$

The above matrix expression 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 formula 10 until convergence.

In summary, the sentence-ranking model for Chinese patents in this paper is defined as the linear combination of heuristic rules and sentence sematic graph, is as follows:

$$SR(S) = \beta * W_{rule}(S) + (1-\beta) * W_{PR}(S) \tag{11}$$

where, β is the weight parameter of the heuristic rules. While $W_{rule}(S)$ is calculated by formula 7 and $W_{PR}(S)$ is calculated by 10.

3) Sentence-Ranking based Keywords Extraction Algorithm: If a sentence contains important information, it and its semantically similar sentences will get higher scores after using the sentence-ranking model. In this way, with the elimination of noise sentences, the effect of keywords extraction from Chinese patents will be greatly improved. However, the sentences of documents are treated equally without considering the semantic importance of different sentences in traditional keywords extraction algorithms. Thus, this paper introduces the semantic weight parameters of sentences produced by sentence-ranking model into the state-of-the-art algorithm TF, so that the semantic importance of sentences can be transferred to the words. TF is defined as follows in this paper.

$$TF(w_i) = \sum_{j=1}^{K_S * n} SR_j * TF_j(w_i) \tag{12}$$

where, w_i is the i-th 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 j-th sentence. $TF_j(w_i)$ is the term frequency of word w_i in the j-th sentence.

There are totally six parameters in our algorithm, called SR based TF-IDF. The damping factor d is generally taken as 0.85 according to PageRank algorithm. The remaining parameters will be discussed next.

IV. Performance Evaluation

A. Datasets and Metrics

Dataset 1. A large amount of original Chinese patent dataset. The original Chinses patents are collected from SIPO during the period from 2016.11.01 to 2016.11.30. We finally accumulated about 1.21 million well-structured Chinese patents, which will be used to, (1) Train a word2vec model for Chinese patents. (2) Generate a IDF dictionary for Chinese patents. (3) As the source of manually annotated Chinese patent corpus.

Dataset 2. A manually annotated patent dataset, which is consist of 557 Chinese patents. This dataset is manually annotated by three masters major in computer science, which has the following requirements: (1) Assign 3-6 keywords to each Chinses patent. (2) Keywords with 2-7 Chinese characters in length. (3) Try to select the word whose POS is noun, verb or adjective.

Finally, we adopt the union of pairwise intersections between the annotations as the human-annotated gold standard dataset for Chinese patents [19].

It can be found from the result of manual annotation that the manually annotated keywords are generally long phrases with specific meanings. In this paper, we primarily focus on the keywords that make up these key phrases. In order to better evaluate the performance of keywords, we consider two forms of agreement:

- 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.

Micro-averaged precision, recall and F-score under these two settings are calculated by the same formula as [19].

B. 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 algorithm. There are five parameters left, including ε, α, β, K_S and K_W Where ε determines the convergence rate of SR based TF-IDF. Eventhough $\varepsilon = 10^{-7}$, the number of iterations still within 20, so ε is taken as 10^{-7} in this paper. α is used to adjust the weights between different heuristic rules, and this paper treats them equally. So that α is taken as 0.5 for the two rules each. The remaining four parameters, β, K_S and K_W, will be discussed one by one in what follows.

In order to discuss the remaining parameter values of β, K_S and K_W, SR based TF-IDF are used to extract keywords from Chinese patents, and we respectively calculate the F-score of Extact-Match and Relaxed-Match, as shown in Table 1 and 2.

The relation between K_W and the optimal F-score is counted as following Table 3 from Table 1 and 2. It is obviously that whatever β and K_S is, the F-score will obtain more optimal values when K_W is 4 (underlined-bold numbers in Table 1 and 2). Therefore, K_W in this paper will be taken as 4 for keywords extraction from Chinese patents.

With the fixed value K_W = 4 and the random value K_S, F-score of Relaxed-Match can get most optimal values only when β = 3 (boxed-underlined-bold numbers in Table 1). However, no matter what the value of β is (3, 4 or 5), they all can get the best F-score of Exact-Match, and F-score gets the most optimal value when β = 5. Thus, to ensure the maximum coverage rate and the minimum average error of optimal F-score in Relaxed-Match and Exact-Match (Table 4), the final value of parameter β is 0.3. Similarly, when K_S = 0.85, F-score of Relaxed-Match and Exact-Match can obtain the global optimal value. On one hand, less noise data can be removed because of the normative and refined contents of Chinese patents, which is consistent with the facts. On the other hand, we can still get better keywords through reducing less noise in Chinese patents. So that the final value of K_S in this paper is 0.85 for keywords extraction from Chinese patents.

Table 1: F-score of Relaxed-Match.

K_S	K_W	β					
		0.1	0.2	**0.3**	0.4	0.5	0.6
0.75	Top3	0.533	0.558	0.568	0.560	0.562	0.561
	Top4	0.561	0.573	**0.588**	**0.577**	**0.575**	**0.570**
	Top5	**0.567**	**0.575**	0.572	0.568	0.562	0.556
	Top6	0.551	0.56	0.558	0.552	0.544	0.535
0.8	Top3	0.528	0.558	0.568	0.558	0.564	0.561
	Top4	0.556	**0.577**	**0.587**	**0.578**	**0.576**	**0.573**
	Top5	**0.562**	0.576	0.571	0.566	0.561	0.556
	Top6	0.551	0.560	0.558	0.552	0.543	0.534
0.85	Top3	0.532	0.553	0.567	0.559	0.564	0.561
	Top4	0.55	**0.577**	**0.589**	**0.583**	**0.576**	**0.573**
	Top5	**0.559**	0.576	0.572	0.565	0.561	0.556
	Top6	0.547	0.557	0.558	0.550	0.543	0.536
0.9	Top3	0.533	0.554	0.566	0.559	0.564	0.561
	Top4	0.548	**0.575**	**0.587**	**0.583**	**0.576**	**0.573**
	Top5	**0.560**	0.574	0.576	0.565	0.559	0.555
	Top6	0.549	0.556	0.559	0.551	0.546	0.537

Table 2: F-score of Exact-Match.

K_S	K_W	β					
		0.1	0.2	**0.3**	0.4	0.5	0.6
0.75	Top3	0.172	0.203	0.208	0.210	0.210	**0.215**
	Top4	0.186	**0.207**	**0.211**	**0.212**	**0.213**	0.211
	Top5	**0.198**	0.202	0.203	0.201	0.199	0.197
	Top6	0.195	0.197	0.197	0.193	0.191	0.186
0.8	Top3	0.171	0.200	0.207	0.208	0.210	**0.215**
	Top4	0.185	**0.207**	**0.210**	**0.211**	**0.213**	0.211
	Top5	**0.196**	0.200	0.201	0.201	0.199	0.197
	Top6	0.195	0.195	0.196	0.193	0.191	0.186
0.85	Top3	0.173	0.197	0.207	0.208	0.212	**0.216**
	Top4	0.182	**0.206**	**0.212**	**0.213**	**0.213**	0.211
	Top5	**0.195**	0.199	0.200	0.201	0.200	0.197
	Top6	**0.195**	0.197	0.194	0.191	0.190	0.186
0.9	Top3	0.172	0.197	0.207	0.208	0.212	**0.216**
	Top4	0.178	**0.203**	**0.211**	**0.213**	**0.213**	0.211
	Top5	0.194	0.198	0.198	0.199	0.198	0.197
	Top6	**0.195**	0.196	0.193	0.191	0.190	0.185

C. Keywords Extraction Results and Discussion

According to section 4.2, the parameters of SR based TF-IDF algorithm have the following values: d = 0.85, $\varepsilon = 10^{-7}$, α = 0.5, β = 0.3, K_S = 0.85 and K_W = 4. In this paper, we compared SR based TF-IDF with a variety of other keywords extraction algorithms, such as TF-IDF, TextRank and the latest word2vec weighted TextRank [20] based on precision, recall and F-score under Exact-Match and Relaxed-Match.

As can be seen from Table 5, the result of the state-of-the-art TF-IDF algorithm and TextRank algorithm is almost the same. The TF-IDF algorithm is simple and effective, and the result is more in line with the actual. However, as the abstracts of Chinese patents are concise, there will be a lot of noise words

Table 3: The relation between K_W and the optimal F-score.

K_W	3	**4**	5	6
the number of optimal F-score	11	**64**	11	1
the proportion of optimal F-score (%)	0.131	**0.762**	0.131	0.012

Table 4: The relation between β and the optimal F-score.

K_W	**0.3**	0.4	0.5
the coverage rate of optimal F-score	**0.643**	0.143	0.357
the average error of optimal F-score (%)	**0.0022**	0.0053	0.0089

Table 5: The performance of keywords extraction algorithm from Chinese patents.

Method	Exact-Match			Relaxed-Match		
	P	R	F	P	R	F
SR based TF-IDF	**0.223**	0.207	**0.212**	**0.623**	**0.571**	**0.589**
TF-IDF	0.087	0.084	0.085	0.515	0.473	0.487
TextRank	0.105	0.125	0.113	0.466	0.533	0.491
word2vec weighted TextRank (2016)	0.121	**0.268**	0.165	0.49	0.549	0.518

with same and low frequency, which cannot be distinguished from the keywords in the state-of-the-art TF-IDF algorithm.

In TextRank, a network topology graph is constructed by the co-occurrence relations between words to get the keywords. However, 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 transfer the weights between words. In order to improve the sparsity of the words graph, a word2vec weighted TextRank [20] is proposed to enrich the semantic relations between words. And the performance is obviously improved from Table 5.

In a word, it is indeed useful for keywords extraction to reduce the noise words and enrich the semantic relationship between words. Then the SR based TF-IDF algorithm proposed in this paper uses a sentence-ranking model to sort the candidate sentences and transfers the semantic weights of sentences into candidate words. Not only consider the semantic relations between words and sentences, the noise of sentences is also reduced. So that the F-score of Exact-Match and Relaxed-Match based on SR based TF-IDF algorithm all achieve the highest score.

V. Conclusions

Keywords extraction from Chinese patents is largely an open problem, with potentially important benefits given the growing number of Chinese patents that we have to handle.

In this paper, we build a sentence-ranking model based on a semantic sentence embedding graph and heuristic rules, and use the model to reduce the noise in Chinese patents. Then the semantic weights of sentences based on the sentence-ranking model are used to calculate the weights of keywords, which make sure that the importance of sentences is transmitted to words. Finally, the experimental results on Chinese patents show that SR based TF-IDF algorithm proposed in this paper improves the performance of keywords by 6% to 13% in F-score, which demonstrated the new idea of selecting key sentences from original documents can effectively filter out noisy sentences and leverage the performance of keywords extraction.

However, the manually annotated patent keywords are always special significate key-phrases. The experiments in this paper concern the keywords that make up those key-phrases, which may cause certain limitations of results. In the future, we will consider the mergence of keywords to get more proper key-phrases to improve the effectiveness of SR based TF-IDF.

Acknowledgment

This research is financially supported by National Natural Science Foundation of China (grant number 61462073) and Science and Technology Committee of Shanghai Municipality (STCSM) (grant number 17DZ1101003).

References

[1] Lai, Chaoan, and X. U. Cuilu. "The Application of Patent Mining in the Forecast of Smart Home Industry." Lancet 381.9876(2016):1458-9.

[2] Fujii, A., et al. "Overview of the patent translation task at the NTCIR-8 workshop." Proc. NTCIR-8 (2010):293-302.

[3] Zhang hongying. "Chinese Key Words Extraction Algorithm." Computer Systems & Applications (2009).

[4] Scientific, Join Faculty Of Computer, et al. "Keyword Extraction Based on New Word Detection." Microcomputer Information (2010).

[5] Csomai, Andras, and R. Mihalcea. "Investigations in Unsupervised Back-of-the-Book Indexing." FLAIRS Conference, 2007:231-42.

[6] You, Wei, D. Fontaine, and J. P. Barthes. "Automatic Keyphrase Extraction with a Refined Candidate Set." Ieee/wic/acm International Conference on Web Intelligence and Intelligent Agent Technology IEEE Computer Society, 2009:576-579.

[7] Beliga, Slobodan, A. Meštrović, and S. Martinčić-Ipšić. "An Overview of Graph-Based Keyword Extraction Methods and Approaches." Journal of Information & Organizational Sciences 39.1(2015):1-20.

[8] Nan, Jiangxia, et al. "Keywords extraction from Chinese document based on complex network theory." ISCID, Vol. 2. IEEE, 2014.

[9] Li, Junfeng, X. Lv, and S. Zhou. "Patent Keyword Indexing Based on Weighted Complex Graph Model." New Technology of Library & Information Service (2015).

[10] Li, Wengen, and J. Zhao. "TextRank Algorithm by Exploiting Wikipedia for Short Text Keywords Extraction." International Conference on Information Science and Control Engineering IEEE, 2016:683-686.

[11] Hasan, Kazi Saidul, and V. Ng. "Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art." International Conference on Computational Linguistics: Posters Association for Computational Linguistics, 2010:365-373.

[12] Zhang, Qingguo, et al. "Automatic Keyword Extraction Based on KNN for Implicit Subject Extraction." Journal of the China Society for Scientific & Technical Information 28.2(2009):163-168.

[13] Cao, Ming, et al. "Comparative research on technology competitiveness based on patent analysis." Studies in Science of Science (2016).

[14] Liu, Dacheng, et al. "Technology Effect Phrase Extraction in Chinese Patent Abstracts." Asia-Pacific Web Conference Springer, Cham, 2014:141-152.

[15] Ding, Wei, Y. Liu, and J. Zhang. "Chinese-keyword fuzzy search and extraction over encrypted patent documents." International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management IEEE, 2015:168-176.

[16] Chen, Yiqun, et al. "Mining Patent Knowledge for Automatic Keyword Extraction." Journal of Computer Research & Development (2016).

[17] Liu, Duan Yang, and L. F. Wang. "Keywords extraction algorithm based on semantic dictionary and lexical chain." Journal of Zhejiang University of Technology (2013).

[18] Juan, Y. U., and Y. Z. Dang. "Chinese term extraction based on POS analysis & string frequency." Systems Engineering-Theory & Practice (2010): 016.

[19] Lahiri, S, R. Mihalcea, and P. H. Lai. "Keyword extraction from emails*." Natural Language Engineering 23.2(2016):295-317.

[20] Wen, Yujun, H. Yuan, and P. Zhang. "Research on keyword extraction based on Word2Vec weighted TextRank." IEEE International Conference on Computer and Communications IEEE, 2017:2109-2113.