Unsupervised Keyword Extraction Method based on Chinese Patent Clustering

Yuxin Xie
Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education
School of Computer Science and Information Engineering
Hefei University of Technology
Hefei, China
xyxhfut@mail.hfut.edu.cn

Xuegang Hu
Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education
School of Computer Science and Information Engineering
Hefei University of Technology
Hefei, China
jsjxhuxg@hfut.edu.cn

Yuhong Zhang
Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education
School of Computer Science and Information Engineering
Hefei University of Technology
Hefei, China
zhangyh@hfut.edu.cn

Shi Li
Intelligent Connected Vehicle Research Institute (JAC Technical Center), Anhui Jianghuai Automobile Group Corp.,Ltd.
Hefei, China
ls.jszx@jac.com.cn

***Abstract*—Recently, patent data analysis has attracted a lot of attention, and patent keyword extraction is a hot problem. Most existing methods for patent keyword extraction are based on the frequency of words without semantic information. In this paper, we propose an Unsupervised Keyword Extraction Method (UKEM) based on Chinese patent clustering. More specifically, we use a Skip-gram model to train word embeddings based on a Chinese patent corpus. Then each patent is represented as a vector called patent vector. These patent vectors are clustered to obtain several cluster centroids. Next, the distance between each word vector in patent abstract and cluster centroid is computed to indicate the semantic importance of this word. The experimental results on several Chinese patent datasets show that the performance of our proposed method is better than several competitive methods.**

***Index Terms*—patent keyword extraction, semantic information, patent clustering, patent vector**

I. Introduction

In 2018, the number of Chinese invention patent applications was grown up to 1.542 million, and the growth rate reached 11.6% according to the statistics of China National Intellectual Property Administration. Many patents have high economic value and are worth being protected in some important fields [1]. Inappropriate analysis of patent technology may lead to patent litigation [2]. Due to the valuable contemporary technological information that patents contained, we can discover the direction of new technologies and predict future trends in related fields by analyzing patent documents [3]. Recently, more and more text mining methods have been utilized to analyze patent documents.

One important research issue in patent analysis is patent keyword extraction. Keywords can be used to summarize the corresponding text and classify documents into categories [4]. However, all Chinese patent documents have no applicant-assigned keywords, which makes artificially assigning keywords for each patent document very difficult. Therefore, we focus on the patent keyword extraction.

Traditional keyword extraction algorithms can be roughly classified into supervised and unsupervised methods. In this paper, we focus on the latter.

In unsupervised keyword extraction methods, most algorithms consider word frequency as a decisive factor in selecting keywords. For example, Term Frequency-Inverse Document Frequency (TF-IDF) was proposed in keyword extraction [5]. However, TF-IDF does not consider the semantic relation between words. TextRank is the typical graph-based method [6], which is similar to the PageRank algorithm. However, due to the high computational complexity, TextRank requires much time to calculate the PageRank values of each word. Rapid Automatic Keyword Extraction (RAKE) [7] can extract keyphrases from texts rapidly, but it cannot extract semantically significant words and the accuracy is not high. Hu et al. [8] proposed the Patent Keyword Extraction Algorithm (PKEA) based on the distributed representation, it has the ability to extract semantic keywords which can provide highly meaningful information from patent documents. However, PKEA needs a large number of pre-defined category words to determine the centroid of each patent category.

In summary, most existing methods for patent keyword extraction are based on the frequency of words without semantic information. In addition, many patent clustering algorithms are based on patent citation networks [9]. In this paper, we propose an Unsupervised Keyword Extraction Method (UKEM) based

on Chinese patent clustering. We consider technical field as the important section for patent clustering. Technical field is a section in patent document that contains multiple categories, and it can indicate the domain involved in the patent. More specifically, firstly, we use Skip-gram model [10] to train word embeddings based on a Chinese patent corpus, and the introduced Skip-gram model can represent the semantic and syntactic relation between words well. Secondly, we calculate the arithmetic mean of the word vectors of each patent technical field, and each vector is used to represent each patent, which called patent vector. Then we use a classical clustering algorithm to cluster related patents automatically by patent vectors, and the centroid of each cluster can be calculated easily. Finally, we predict the relevant cluster for each test patent abstract, and then the distance between each candidate word vector and the relevant cluster centroid is designed to indicate the importance of candidate words individually. The experimental results demonstrate the effectiveness of our method.

Our main contributions in this paper can be summarized as follows:

- We propose an unsupervised keyword extraction method based on Chinese patent clustering, which contains semantic information. In our method, we consider technical field as the important section for patent clustering.
- We release a manually tagged patent keyword dataset, consisting of 600 Chinese patent documents with high-quality manually tagged keywords in several fields. This dataset is accessible in our github repository[1].

The rest of the paper is organized as follows. Section II discusses related works in keyword extraction research. Section III provides a detailed description of our proposed algorithm. Section IV shows our experimental results. Section V summarizes this paper and discusses future work.

II. Related Work

Keyword extraction has been playing an important role in classic text mining tasks for many years, which is the basis for text retrieval and text classification. In the field of patents, keyword extraction has received a lot of attention as an important way to obtain advanced technology from patent documents [11]. Kim, et al. [12] built a semantic network of keywords from patent documents in order to predict emerging technologies. Lee, et al. [13] found that new technological opportunities can be identified by building a patent keyword evolution map. However, all patent documents have no applicant-annotated keywords. Thus, many researchers have focused on patent keyword extraction.

The first step in extracting keywords from patents is to decide the most appropriate section of the patent documentation for keyword extraction. Patent documents are divided into multiple sections, including title, abstract, claims, and description. The number of words in these sections is quite different from each other because of their different purposes. Noh, et al. [14] proposed guidelines for extracting keywords. They evaluated the keyword extraction performance considering different sections of patents. They found that the most suitable keyword extraction strategy for patent research is selecting words from the abstract section.

The second phase is to determine the keyword extraction algorithm. Generally, based on whether a labeled corpus is required, existing keyword extraction algorithms can be roughly classified into supervised and unsupervised methods.

In supervised keyword extraction methods, a large number of tagged keywords are required for training a classifier. This classifier determines which words in the document are important. Many commonly used classification algorithms have been tried. Ardiansyah et al. [15] extracted valuable information from trained neural networks by using decision tree. Domoto et al. [16] proposed a spoken term detection method using SVM-based classifier. This SVM-based classifier was trained with a large number of pre-indexed keywords. Yu et al. [17] proposed a new Hidden-State Maximum Entropy (HSME) model for estimating word confidence. Their method outperforms decision tree and SVM in most keyword-spotting tasks. Liu et al. [18] proposed a semi-automatic method based on partitioning corpus to extract technology effect phrases in Chinese patent abstracts. Their method needs a few manually extracted patent abstracts to construct a domain-independent corpus.

Unsupervised keyword extraction methods can be divided into statistical methods, neural network based methods, and graph-based methods. Nam et al. [19] proposed a keyword based method that can monitor the adoption of existing technology by term frequency-inverse document frequency (TF-IDF) and K-means clustering using cited patents. When the number of patents using TF-IDF and K-means clustering was too small, they considered patents that were cited by the originally set of patents. This method is suitable for US patents but not for Chinese patents because Chinese patents have no cited patents. Kim et al. [20] proposed a new method using a paper to extract keywords with unique characteristics of a single patent. They used a frequency analysis and a co-word analysis to improve the performance of keywords extraction. Wu et al. [21] proposed an unsupervised neural network based keyword extraction algorithm, which was inspired by the visual attention mechanism. Their algorithm can effectively extract keywords with rich contextual information from the document. Wang et al. [22] proposed a sentence-ranking model based on a sentence embedding graph and heuristic rules. Their experimental results show that keywords are likely to exist in key sentences. They believe that the importance of a sentence is related to the position of the sentence, and the importance of a word is related to the part of speech and the function of the word. Wen et al. [23] proposed a keywords graph model based on the basic idea of TextRank. They used Word2Vec to calculate the similarity between words as transition probability of nodes' weight. Li et al. [24] proposed a graph-based ranking algorithm by exploiting Wikipedia as an external knowledge base for short text keywords extraction.

[1]https://github.com/NoteXYX/Manually-tagged-patent-keyword-dataset

They pointed out that the Wikipedia's concept can be used to enrich the semantic information of each word in TextRank algorithm.

In summary, the research of keyword extraction has made great progress. We are aware that patent keyword extraction has great influence on patent analysis and becomes a hot problem. However, there are only a little previous studies about keyword extraction algorithm for Chinese patents [22]. Therefore, we introduce a novel keyword extraction method for Chinese patents, and obtain the better performance than several competitive methods.

III. Methodology

A. Overview of the Framework

The framework of our proposed UKEM is shown in Fig. 1, which contains a preprocessing module, a clustering module and a keyword extraction module. These modules will be introduced in detail next.

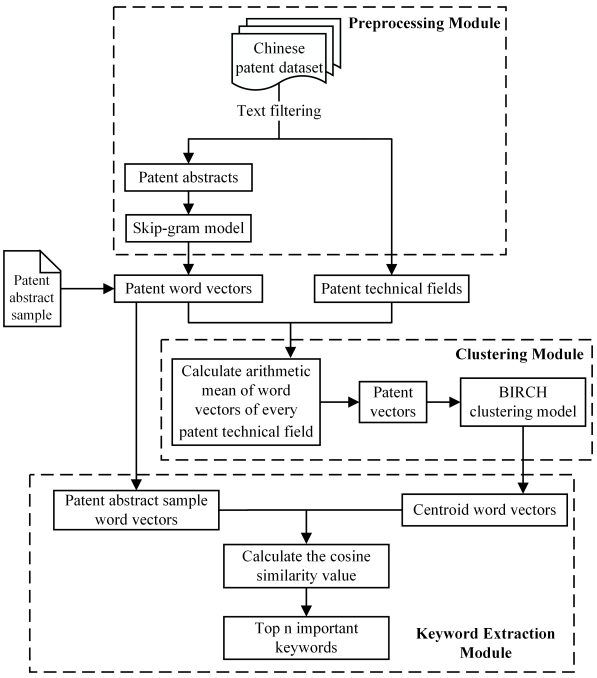

Fig. 1. The framework of our proposed UKEM.

B. Preprocessing Module

We use a Chinese patent corpus to construct a Chinese patent database. Chinese patent documents have strict and standardized templates and writing criterions. Most previous studies have extracted keywords from patent abstracts or patent titles. In this paper, we consider abstract and technical field as two important sections in patent keyword extraction.

Firstly, we extract abstract from each patent document in our Chinese patent database. We use jieba[2] as the Chinese word segmentation tool. However, after the procedure of word segmentation, a large number of meaningless words and punctuations may be produced. Therefore, we create a stop words table to filter those meaningless words and punctuations in extracted abstracts.

Secondly, because the patent abstract has hundreds of words and clearly states the technical problems to be solved and the primary technical method adopted by the patent, the words in the patent abstract are closely related. Therefore, the abstracts of patents are used as a corpus to train Skip-gram model. Skip-gram model is a distributed word representation approach based on a neural network, and it can encode the semantic and syntactic information in to real-valued, dense and low-dimensional vectors. This model has the ability to learn high-quality word vectors from unstructured text data with billions of words and construct a large vocabulary. After the training procedure, each word in patent abstracts obtains the corresponding word vector which can be considered as the projection of the word in a semantic and syntactic space.

Finally, according to IPC (International Patent Classification) code, we extract the technical fields of the three categories of patents and also filter stop words which are meaningless. IPC code consists of sections, classes, subclasses, main group, and sub-groups, it directly reflects the technical topic of the patent. In this paper, we use 3-digit IPC code to judge the category of a patent. Then we remove few technical fields whose words do not appear in any patent abstracts.

C. Clustering Module

After executing the preprocessing module, we obtain all word vectors in patent abstracts and the technical fields of the three categories of patents. The technical field of a Chinese patent usually only contains one sentence. We calculate the arithmetic mean of the word vectors of each patent technical field in three categories of patents, and the result is called patent vector. The dimension of patent vector is the same as that of word vector. The formula of patent vector is defined as:

$$V_{p_i} = \frac{1}{n_i} \sum_{j=1}^{n_i} V_{w_{ij}} \tag{1}$$

where V_{p_i} is the patent vector of the i-th patent p_i and n_i is the total number of words in the technical field of the patent p_i. w_{ij} represents the j-th word in the technical field of the patent p_i and $V_{w_{ij}}$ is its word vector produced by Skip-gram model.

Then we use a classical clustering algorithm, Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) [25], to cluster related patents automatically based on patent vectors. BIRCH is a scalable clustering method designed for very large data sets, and it is a clustering method that does not specify the number of categories.

After the BIRCH clustering procedure, we can calculate the centroid of each patent cluster. The formula of the centroid of each patent cluster is defined as:

$$C_i = \frac{1}{m_i} \sum_{j=1}^{m_i} V_{p_{ij}} \tag{2}$$

[2]https://github.com/fxsjy/jieba

where C_i is the centroid vector of the i-th patent cluster, and m_i is the total number of patents in the i-th patent cluster, and $V_{p_{ij}}$ is the j-th patent vector in the i-th patent cluster.

D. Keyword Extraction Module

As mentioned above, we already have trained the patent word vectors and generated the centroid vector for each patent cluster. The patent document sample that needs to extract keywords must belong to one of the above patent clusters. Keywords in the patent sample document will be extracted from the abstract.

Firstly, we keep the candidate words in patent sample documents with length from 2 to 6. Each candidate word in the abstract of the patent sample document is converted to a word vector according to the patent word vectors. We use BIRCH clustering model trained in clustering module to predict the category of the patent sample document. Then the relevant cluster centroid is obtained.

Secondly, the distance between the candidate word vector and the cluster centroid is calculated by using cosine similarity. The cosine similarity is expressed as:

$$Sim(V_{w_{ij}}, C_k) = \cos(V_{w_{ij}}, C_k) = \frac{V_{w_{ij}} \bullet C_k}{\|V_{w_{ij}}\| \|C_k\|} \quad (3)$$

where w_{ij} is the j-th word in the abstract of the i-th patent sample document, and C_k is the centroid vector of the k-th patent cluster. $V_{w_{ij}}$ represents the word vector of w_{ij}.

The importance of a word in the abstract of the patent sample document is related to the distance between the candidate word vector and the cluster centroid. The closer the word vector is to the clustering centroid, the more important the word is.

Finally, the cosine similarity values are sorted from the largest to the smallest. The extracted keywords for each document are the top n words whose word vectors have the largest similarity values with relevant cluster centroid.

There are several parameters of the Skip-gram model and the BIRCH clustering model in our method. These parameters will be discussed next.

IV. Experiments

In this section, we conduct several experiments to demonstrate that our method can extract keywords from Chinese patents more accurately than several competitive methods.

In the following, we firstly describe the details of patent datasets and evaluation metrics used in our experiments. Secondly, we discuss about parameters setting in our method. Then the benchmark methods are introduced in detail. Finally, the experiment results of patent clustering and patent keyword extraction are given, and we will analyze these results.

A. Datasets and Evaluation Metrics

We finally accumulated 263,143 well-structured Chinese patents from the China National Intellectual Property Administration and constructed a Chinese patent database. These patents cover a lot of fields and their application dates are from 2012 to 2018. All Chinese patent documents are divided into multiple sections, including patent number, IPC code, title, inventor, abstract, technical field, background technology, and invention content. Because of their different purposes, their sentence structures are different from each other. We extract abstracts from all patent documents as the corpus for training the patent word embeddings. An example of a Chinese patent is shown in Fig. 2. In our experiment, we only use title, abstract and technical field in patent documents.

According to 3-digit IPC code, we extract six categories of patent documents to construct a patent clustering dataset. Table I shows the 3-digit IPC code, the category name and the number of patents in each category.

TABLE I
PATENT CLUSTERING DATASET

IPC code	***Information***	
	Category Name	***Number of Patents***
F25D	Refrigerator	539
D06F	Textile washing	799
H04M	Telephone communication	998
F24F	Air conditioning	1,000
H04N	Television	1,000
B08B	Cleaning	687

In order to test the performance of our algorithm, we extract 100 patents from each category in the patent clustering dataset and construct a manually tagged patent keywords dataset. The total number of patents in this patent keywords dataset is 600. The keywords in this dataset are manually tagged by three masters major in patent. We give these masters two requirements. The first requirement is to tag 5-10 keywords for each Chinese patent in patent keywords dataset, ranked

Title: 空调器湿度控制方法及空调器
Abstract: 本发明公开一种空调器湿度控制方法及空调器，其中，空调器湿度控制方法包括以下步骤：判断是否满足加湿条件；当满足加湿条件时，控制空调器进行加湿，并对加湿时间进行累计；当累计的加湿时间达到第一设定时间时，控制空调器停止加湿，并对加湿停止时间进行累计；当累计的加湿停止时间达到第二设定时间时，返回执行所述判断是否满足加湿条件的步骤。本发明技术方案可避免墙壁凝露现象的产生。
Technical Field: 本发明涉及空调技术领域，特别涉及一种空调器湿度控制方法及空调器。

Fig. 2. An example of a Chinese patent document in which we only use title, abstract, and technical field.

with their importance. The second requirement is to keep the tagged keywords with 2-6 Chinese characters in length. Then we adopt the union of annotations as the human-tagged gold standard keywords dataset for Chinese patents [26].

We calculate $Precision_{score}$, $Recall_{score}$ and $F1_{score}$ for each keyword extraction. The precision score is defined as:

$$Precision_{score} = \frac{W_{ca}}{topk_a} \tag{4}$$

where Wca is the number of correct keywords extracted by algorithm, and $topk_a$ is the number of top k keywords extracted by algorithm. The recall score is defined as:

$$Recall_{score} = \frac{W_{ca}}{topk_h} \tag{5}$$

where Wca is the number of correct keywords extracted by algorithm, and $topk_h$ is the number of top k keywords in the human-tagged gold standard. The F1 score is defined as:

$$F1_{score} = \frac{2 \times Precision_{score} \times Recall_{score}}{Precision_{score} + Recall_{score}} \tag{6}$$

We use F1 score to test the performance of our algorithm and several competitive algorithms.

B. Compared Methods

We compare our proposed UKEM with the following baseline methods:

- Rapid Automatic Keyword Extraction (RAKE) [7]. It can extract key phrases rapidly from texts. The computational complexity of this algorithm is very low, so it is suitable for keyword extraction tasks on large-scale text.
- Term Frequency-Inverse Document Frequency (TF-IDF) [5]. It is based on statistical methods, and it is the most widely applied keyword extraction algorithm. This algorithm considers word frequency as the decisive factor in extracting keywords.
- TextRank [6], which is the classical graph-based keyword extraction algorithm. This algorithm requires a lot of iterations to calculate the PageRank values for each word, so it is suitable for keyword extraction tasks on small-scale text.
- Patent Keyword Extraction Algorithm (PKEA) [8], which is specially designed for extracting keywords from patent documents. This algorithm requires pre-defined category corpus to describe the technical topic of each category of patents. Skip-gram model is been used in this algorithm, so it contains semantic information.

C. Parameters Selection

In preprocessing module, there are three parameters that need to be set before training the Skip-gram model. The minimum word count is set to 1, because we think that some words describing emerging technology fields may only appear once, but they are still important. The window size and the dimension of patent word vector is set to 10 and 100 respectively.

In clustering module, there are three parameters that need to be set before training the BIRCH clustering model. The first parameter is the maximum CF number B of each internal node in CF tree, it is usually taken as 50. The second parameter is the maximum CF number L of each leaf node in CF tree. Generally, L is set the same as B, so L is also set as 50. The third parameter is the maximum sample radius threshold T. To get the best performance of patent clustering, we set T as 1.04 for the first group of patent clustering dataset which consists of refrigerator, textile washing and telephone communication patents, 1.0115 for the second group of patent clustering dataset which consists of air conditioning, television and cleaning patents, 1.006 for the third group of patent clustering dataset which consists of refrigerator, textile washing and air conditioning patents.

There is an important damping factor in TextRank, it is generally taken as 0.85 according to PageRank algorithm.

D. Experiment Results and Discussion

We randomly select three groups of patents from the patent clustering dataset for patent clustering. Each group has three

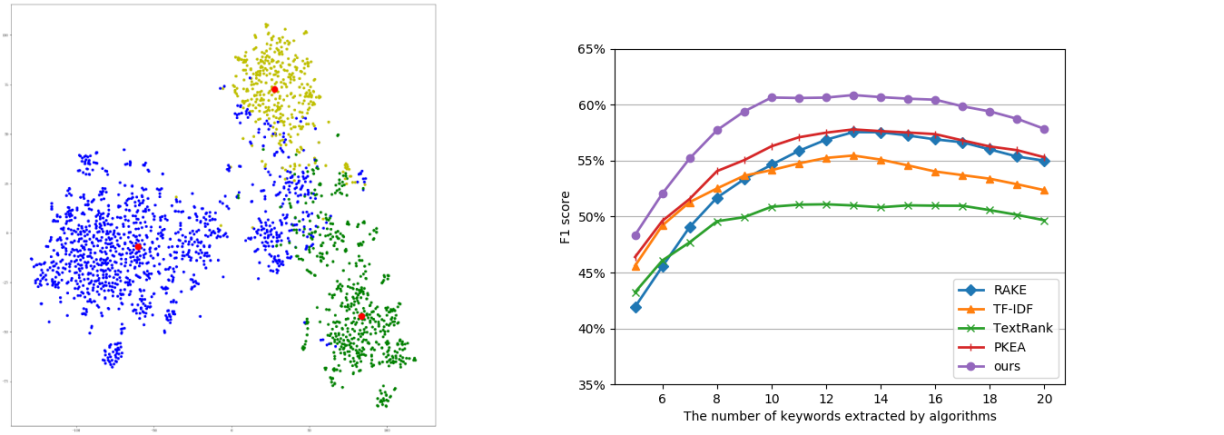

Fig. 3. The schematic diagram of clustering result on the first group of patent clustering dataset and F1 scores on extracting patent keywords from 100 patents in each patent cluster.

categories of patents. The first group of patents consists of 2,336 patents, the categories of these patents include refrigerator, textile washing and telephone communication. The second group of patents consists of 2,687 patents, the categories of these patents include air conditioning, television and cleaning. The third group of patents consists of 2,338 patents, the categories of these patents include refrigerator, textile washing and air conditioning.

Then we extract keywords from 100 patents in each category of patents by algorithms and calculate the F1 score on matching keywords extracted by algorithms with manually tagged ones. These patents are in our manually tagged patent keywords dataset.

Fig. 3-5 show the clustering results on the three groups of patents and the F1 scores on matching keywords extracted by algorithms with manually tagged ones respectively. All points in blue, green and yellow represent patent vectors. The dimension of patent vector is 100, the same as that of patent word vector. We use t-SNE [27] to reduce the dimension of patent vectors and display them. Points with the same color belong to the same cluster.

The red point is the centroid of cluster. In order to analyze the impact of different numbers of keywords on patent keyword extraction, the number of keywords extracted by algorithms is set from 5 to 20.

We have the following observations from above experimental results:

- When the number of extracted keywords is 13, almost all algorithms get the highest F1 score.
- The performance of our method is better than several competitive algorithms. The highest F1 score among all algorithms on each group of patents is obtained by our proposed method. When the number of extracted keywords is 13, the F1 scores of our method are improved by 1%-12% compared to other algorithms. Specifically, compared to RAKE, TF-IDF and TextRank, the F1 scores of PKEA and our method are improved by 0.2%-12%, it indicates that semantic information is effective for patent keyword extraction. Compared to PKEA, the F1 scores of our method are improved by 1%-4%, it indicates that our method can find the cluster centroid more accurately.
- It is worth noting that the clustering result on the third

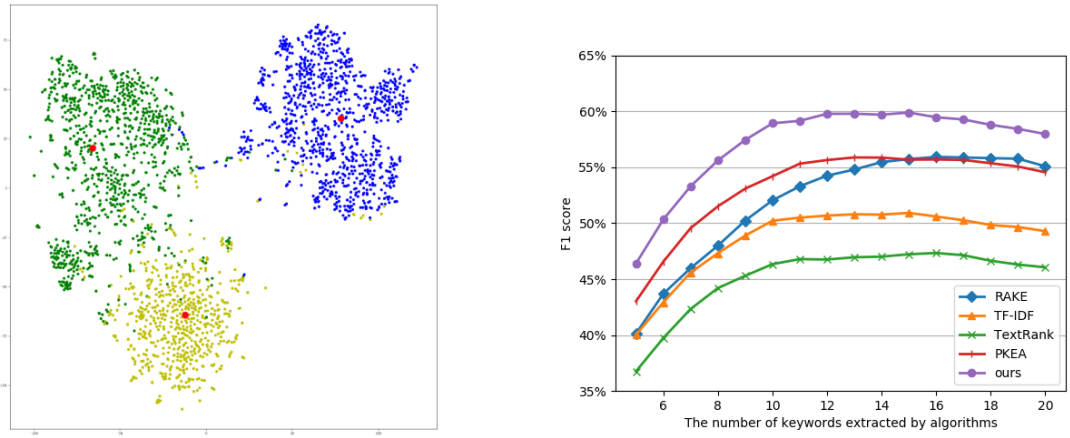

Fig. 4. The schematic diagram of clustering result on the second group of patent clustering dataset and F1 scores on extracting patent keywords from 100 patents in each patent cluster.

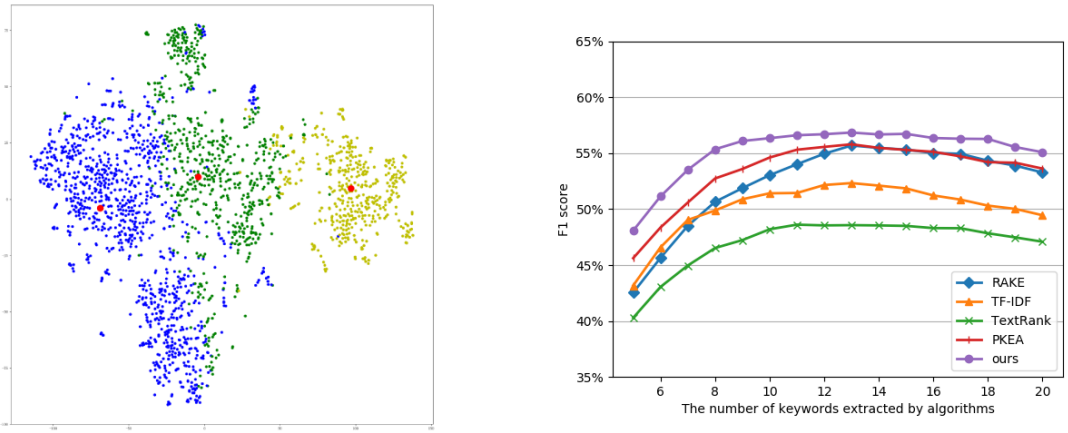

Fig. 5. The schematic diagram of clustering result on the third group of patent clustering dataset and F1 scores on extracting patent keywords from 100 patents in each patent cluster.

group of patents is not as good as the previous two groups of patents. Because refrigerator patents and air conditioning patents (blue points and green points) both use similar technical methods in some technical fields such as refrigerants, compressors, and temperature regulation, our method cannot accurately distinguish them in patent vector space. But our method can distinguish textile washing patents (yellow points) accurately.

In order to consider the impact of constructing patent vectors using different sections in patents on the results of patent keyword extraction, we use abstracts, titles and technical fields of patents to construct patent vectors respectively for clustering. When the number of extracted keywords is 13, the F1 scores on patent keyword extraction which use three different sections in patents to construct patent vectors are shown in Fig. 6. In Group 1 and Group 2 patent clustering data, the F1 scores on patent keyword extraction using technical field to construct patent vectors are obviously higher than using abstract and title. Therefore, it is effective on constructing patent vectors using technical field. In Group 3 patent clustering data, as described above, the clustering performance is not good, so the result on patent keyword extraction which use technical field to construct patent vectors is not good as Group 1 and Group 2.

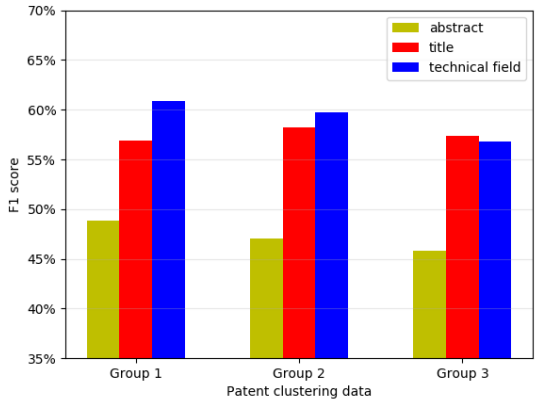

Fig. 6. F1 scores on patent keyword extraction which use three different sections in the three groups of patents to construct patent vectors when the number of extracted keywords is 13.

In summary, it is indeed useful for patent keyword extraction to cluster patents which use similar technical methods and enrich the semantic information between words.

V. Conclusion and Future Work

The patent document potentially contains many innovative technical methods. Patent keywords can help us discover the technology used in patents more efficiently.

In this paper, we propose an unsupervised patent keyword extraction method based on Chinese patent clustering. We use the arithmetic mean of the word vectors of each patent technical field as the patent vector to represent each patent. Then we use BIRCH clustering model to cluster several patent vectors. The centroid of each patent cluster can be calculated easily. Next, the cosine similarity between each word vector in patent abstract and cluster centroid is computed to indicate the importance of this word. Finally, the extracted keywords for each patent document are the top n words whose word vectors have the largest similarity values with relevant cluster centroid.

However, the clustering results on the patents which use similar technical methods in some technical fields are not so good. Our future work is to cluster patents on a certain category of patents, each cluster contains the patents which use the same specific technology. Moreover, we plan to add position information of each word for patent keyword extraction.

Acknowledgment

This work is supported in part by the Natural Science Foundation of China under grants (61673152, 61976077, 2016YFC0801406).

References

[1] L. Tahmooresnejad and C. Beaudry, "Capturing the economic value of triadic patents," *Scientometrics*, vol. 118, no. 1, pp. 127–157, 2019.

[2] Q. Liu, H. Wu, Y. Ye, H. Zhao, C. Liu, and D. Du, "Patent litigation prediction: A convolutional tensor factorization approach." in *IJCAI*, 2018, pp. 5052–5059.

[3] S. Lee, B. Yoon, and Y. Park, "An approach to discovering new technology opportunities: Keyword-based patent map approach," *Technovation*, vol. 29, no. 6-7, pp. 481–497, 2009.

[4] A. Onan, S. Korukoğlu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Systems with Applications*, vol. 57, pp. 232–247, 2016.

[5] S. Lee and H.-j. Kim, "News keyword extraction for topic tracking," in *2008 Fourth International Conference on Networked Computing and Advanced Information Management*, vol. 2. IEEE, 2008, pp. 554–559.

[6] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004.

[7] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents," *Text mining: applications and theory*, pp. 1–20, 2010.

[8] J. Hu, S. Li, Y. Yao, L. Yu, G. Yang, and J. Hu, "Patent keyword extraction algorithm based on distributed representation for patent classification," *Entropy*, vol. 20, no. 2, p. 104, 2018.

[9] A. Rodriguez, A. Tosyali, B. Kim, J. Choi, J.-M. Lee, B.-Y. Coh, and M. K. Jeong, "Patent clustering and outlier ranking methodologies for attributed patent citation networks for technology opportunity discovery," *IEEE Transactions on Engineering Management*, vol. 63, no. 4, pp. 426–437, 2016.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[11] Y.-R. Li, L.-H. Wang, and C.-F. Hong, "Extracting the significant-rare keywords for patent analysis," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5200–5204, 2009.

[12] Y. G. Kim, J. H. Suh, and S. C. Park, "Visualization of patent analysis for emerging technology," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1804–1812, 2008.

[13] S. Lee, H.-j. Lee, and B. Yoon, "Modeling and analyzing technology innovation in the energy sector: Patent-based hmm approach," *Computers & Industrial Engineering*, vol. 63, no. 3, pp. 564–577, 2012.

[14] H. Noh, Y. Jo, and S. Lee, "Keyword selection and processing strategy for applying text mining to patent analysis," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4348–4360, 2015.

[15] S. Ardiansyah, M. A. Majid, and J. M. Zain, "Knowledge of extraction from trained neural network by using decision tree," in *2016 2nd International Conference on Science in Information Technology (ICSITech)*. IEEE, 2016, pp. 220–225.

[16] K. Domoto, T. Utsuro, N. Sawada, and H. Nishizaki, "Spoken term detection using svm-based classifier trained with pre-indexed keywords," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 10, pp. 2528–2538, 2016.

[17] P. Yu, J. Xu, G.-L. Zhang, Y.-C. Chang, and F. Seide, "A hidden-state maximum entropy model forword confidence estimation," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 4. IEEE, 2007, pp. IV–785.

[18] D. Liu, Z. Peng, B. Liu, X. Chen, and Y. Guo, "Technology effect phrase extraction in chinese patent abstracts," in *Asia-Pacific Web Conference*. Springer, 2014, pp. 141–152.

[19] S. Nam and K. Kim, "Monitoring newly adopted technologies using keyword based analysis of cited patents," *IEEE Access*, vol. 5, pp. 23 086–23 091, 2017.

[20] J. Kim, J. Choi, S. Park, and D. Jang, "Patent keyword extraction for sustainable technology management," *Sustainability*, vol. 10, no. 4, p. 1287, 2018.

[21] X. Wu, Z. Du, and Y. Guo, "A visual attention-based keyword extraction for document classification," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 25 355–25 367, 2018.

[22] Z. Wang, Y. Guo, and T. Qi, "Keywords extraction based on sentence-ranking from chinese patents." in *SEKE*, 2018, pp. 80–85.

[23] Y. Wen, H. Yuan, and P. Zhang, "Research on keyword extraction based on word2vec weighted textrank," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2016, pp. 2109–2113.

[24] W. Li and J. Zhao, "Textrank algorithm by exploiting wikipedia for short text keywords extraction," in *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*. IEEE, 2016, pp. 683–686.

[25] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: an efficient data clustering method for very large databases," in *ACM Sigmod Record*, vol. 25, no. 2. ACM, 1996, pp. 103–114.

[26] S. Lahiri, R. Mihalcea, and P.-H. Lai, "Keyword extraction from emails," *Natural Language Engineering*, vol. 23, no. 2, pp. 295–317, 2017.

[27] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.