Knowl Inf Syst (2018) 57:135–158
https://doi.org/10.1007/s10115-017-1127-0

REGULAR PAPER

PatSearch: an integrated framework for patentability retrieval

Longhui Zhang[1,2] · Zheng Liu[1] · Lei Li[2] · Chao Shen[2] · Tao Li[1,2]

Received: 3 May 2016 / Revised: 4 September 2017 / Accepted: 23 October 2017 /
Published online: 6 November 2017

Abstract Patent retrieval primarily focuses on searching relevant legal documents with respect to a given query. Depending on the purposes of specific retrieval tasks, processes of patent retrieval may differ significantly. Given a patent application, it is challenging to determine its patentability, i.e., to decide whether a similar invention has been published. Therefore, it is more important to retrieve all possible relevant documents rather than only a small subset of patents from the top ranked results. However, patents are often lengthy and rich in technical terms. It is thus often requiring enormous human efforts to compare a given patent application with retrieved results. To this end, we propose an integrated framework, `PatSearch`, which automatically transforms the patent application into a reasonable and effective search query. The proposed framework first extracts representative yet distinguishable terms from a given application to generate an initial search query and then expands the query by combining content proximity with topic relevance. Further, a list of relevant patent documents will be retrieved based on the generated queries to provide enough information to assist patent analysts in making the patentability decision. Finally, a comparative summary is generated to assist patent analysts in quickly reviewing retrieved results related to the patent application. Extensive quantitative analysis and case studies on real-world patent documents demonstrate the effectiveness of our proposed approach.

Keywords Patent retrieval · Query extraction · Query expansion · Knowledge base

✉ Tao Li
taoli@cs.fiu.edu

[1] Jiangsu BDSIP Key Lab, School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, People's Republic of China

[2] School of Computing and Information Sciences, Florida International University, Miami, FL 33199, USA

1 Introduction

Among intellectual resources, one important type is patent documents, which help protect interests of companies and organizations. Different from general Web documents (e.g., Web pages), the format of patent documents is well defined. They are lengthy and rich in technical terms and hence often require intensive human efforts for analysis. Therefore, patent retrieval, aiming to assist patent analysts in retrieving, processing and analyzing patent documents, emerges as a new popular research area in recent years [16,17,41].

Patent retrieval tasks in practice may have various purposes. Typical patent retrieval tasks [16] include *prior-art search* (understanding the state of the art of a targeted technology), *patentability search* (retrieving relevant patent documents to check if similar ideas exist), *infringement search* (examining if a product infringes a valid patent or not). Due to the great commercial value of patents and significant costs of processing a patent application (or a patent infringement case), the common requirement of these patent retrieval tasks is to providing full coverage with respect to the query document as much as possible.

The high quality of search queries is the cornerstones of patent retrieval; however, it is not a trivial task to build/find such queries. In order to ensure the patentability of patent documents and maximize the scope of the protection, patent inventors and attorneys usually use complex sentences containing domain-specific words to describe the invention. This renders patent documents, especially the patent claims (which define the implementation of essential components of patent inventions), difficult to read through and understand. A common practice of generating the expected query is to manually extract representative terms from original patent documents by domain expects. This process often requires a tremendous amount of time and human efforts. Hence, it is imperative to automate this process, which will assist the analysts in finding relevant patent documents conveniently. As an example, Xue et al. [39] extracted query terms from the summary field of a patent document and relied on the term frequency to automatically transform a patent file into a query.

On the other hand, patentability retrieval is a recall-orientated search task. Missing relevant patent documents is not allowed in patentability retrieval because of the high commercial value of patents and high costs of processing a patent application or patent infringement. Thus, it is important to retrieve all possible relevant documents rather than finding only a small subset of relevant patents from the top ranked results. To this end, a common practice is to enrich the query keywords in order to improve the keyword coverage, which is often referred to as *query expansion*. Recently, many query expansion techniques have been introduced in the field of patent search to improve the effectiveness [19,22,25]. However, despite recent advances of query expansion, several critical issues in the current generation of patent search systems have not been well explored in previous studies. For example, the expansion of query terms may result in topic drift, i.e., the topics of the query may change/shift to an unintended direction after query expansion. Another critical issue is the ambiguity of a search query, i.e., a single term may have multiple meanings with respect to specific contexts.

Finally, even for only a few retrieved patent documents, it is not trivial to analyze the results. For instance, the task of determining patentability involves analyzing previous patent documents that possibly disclosed the content of the filing patent application. Analysts have to read through all the retrieved patent documents to determine whether the filing patent application satisfied the patentability requirements. Nonetheless, patent application documents are often lengthy, and full of technical and legal terminologies. Even for domain experts, reading and analyzing a single patent document might take a fairly long time. Hence, it is imperative to assist the analysts in efficiently reviewing the relationship between the query and the

retrieved patents. Some recent studies advance patent retrieval technologies [1,2,7,30], but the comparison process is still far from being well addressed in both research and industry communities. In addition, to the best of our knowledge, our work is one of the first research studies toward reducing human efforts of comparing patent documents by leveraging comparative summarization techniques.

To address the aforementioned issues, we propose a unified framework, named `PatSearch`, wherein a user submits the entire patent application as the query. Given a patent application, `PatSearch` will automatically extract representative yet distinguishable terms to generate a search query. In order to alleviate the issues of ambiguity and topic drift, a new query expansion approach is proposed, which combines content proximity with topic relevance. Further, the system automatically compares the retrieved patent documents with the given query and generates a comparison report per user request. `PatSearch` aims to help users retrieve relevant patent documents as many as possible and provide enough information to assist patent analysts in making the patentability decision.

Specifically, our proposed framework has the following significant merits:

- *Automatic keywords extraction*: Based on the analysis of patent documents, `PatSearch` is able to automatically extract important yet distinguishable keywords from a given patent application, which integrates special characters of patent documents (e.g., patent classification code and patent structure).
- *Relevant query expansion*: Based on the knowledge base and term thesaurus, `PatSearch` is capable of expanding a list of keywords related to a given query term. The expansion is achieved by combining the content proximity with topic relevance.
- *Comparative summary generation*: `PatSearch` utilizes graph-based techniques to build connections of two patent documents from various aspects, resulting in a term co-occurrence graph, and automatically generate comparative summaries. Such summaries are able to help patent analysts quickly go through the retrieved results.

The rest of the paper is organized as follows: Sect. 2 introduces the background and reviews the related work; Sect. 3 presents the `PatSearch` framework; Sect. 4 presents the detailed technical approaches used in `PatSearch` for retrieving relevant patents; Sect. 5 provides the techniques used for comparing two patent documents; Sect. 6 shows the experimental results and present case studies; and finally Sect. 7 concludes the paper.

2 Background and related work

2.1 Definition of patentability

Patents serve to protect the intellectual properties of patent owners. Patent laws and regulations are often different in different countries and regions, but in general, the requirements of patentability of inventions are similar. According to the US patent law,[1] US Code Title 35, § 102(f),"*Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.*" An utility invention is patentable in US, if and only if it is:

- patentable subject matter, i.e., a type of subject matters eligible for patent protection (e.g., process, machine, manufacture)

[1] https://www.uspto.gov/web/offices/pac/mpep/.

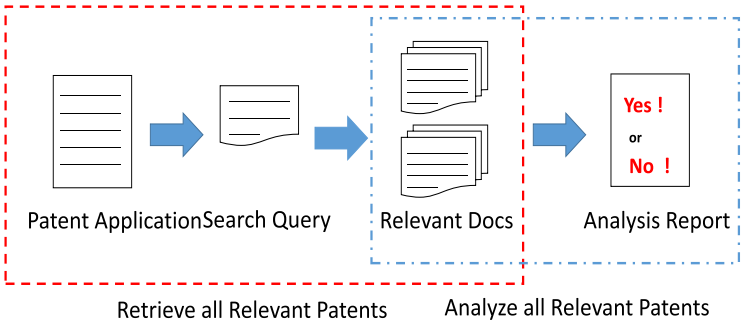


Fig. 1 A workflow of patentability search

- novel, i.e., the same invention has not been disclosed before the effective filing date;
- non-obvious, i.e., a "person having ordinary skill in the art" would not know how to solve the problem using the same mechanism;
- useful, i.e., it provides some identifiable benefit and is capable of use.

Details on patentability in the US can be found in the Manual of Patent Examining Procedure(MPEP),[2] Chapter 2100. In a typical patentability retrieval task, patent analysts put more attention on verifying whether an invention satisfies the requirements and conditions of novelty and non-obvious nature. However, it is not easy to decide whether the filing patent application is non-obvious or not as it requires to apply the test for obviousness. In the test, finding the differences between the filing patent application and relevant prior arts is the major task.

2.2 Process of patentability retrieval

Patentability retrieval is a subfield of patent retrieval, in which the basic search element is a patent document. Due to the characteristics of patent documents and special requirements of patent retrieval, patentability search is quite different from searching general Web documents. For example, queries in patentability search are generally much longer and more complex than the ones in Web search. A simplified workflow of patentability search is shown in Fig. 1.

In general, the process of patentability search includes two stages: In Stage 1, given a patent application, patent analysts need to retrieve all the relevant patents in order to determine the patentability, and in Stage 2, patent analysts need to review all the retrieved patents to make the decision. Despite the recent advances, the task of patentability search remains challenging from multiple perspectives, such as low readability, lengthy query and high recall demand.

2.2.1 Patent query generation

In general, users may specify only several keywords in ad hoc Web search. Most Web-based search systems restrict the length of input queries, e.g., the maximum number of query keywords in Google search engine is 32. One possible reason is that the retrieval response time of search engines increases along with the length of the input. Comparatively, a patent query often consists of tens or even hundreds of keywords on average in patent retrieval systems. A common practice of generating such a query is to manually extract representative terms

[2] https://www.uspto.gov/patent/laws-regulations-policies-procedures-guidance-and-training/.

from original patent documents or add additional technological terms. This is often achieved by patent examiners, which requires a tremendous amount of time and human efforts. In addition, patent examiners are expected to have strong technological background in order to provide a concise yet precise query. To assist patent examiners in generating patent queries, a lot of research work has been proposed in the last decade.

Query extraction aims to extract representative information from an invention that describes the core idea of the invention. The simplest way of query extraction is to use the abstract, the summary of the invention, or the independent claims, the protection scope of the invention. However, the extracted information based on abstracts or claims may not be suitable to form the patent query. The reason is obvious: Applicants often describe the abstract/claim without enough technical details in order to decrease the retrievability of their patent, and the terms in the abstract/claims often contain obscure meaning (e.g., "comprises" means "consists at least of") [33]. Additional efforts along this direction often involve extracting query terms from different patent document sections to automatically transform a patent file into a query [24,39]. In Xue and Croft [39], different weights are assigned to terms from different sections of patents. Their experiments on a USPTO patent collection indicate that using the terms from the description section can produce high-quality queries, and using the term frequency weighting scheme can achieve superior retrieval performance. In Mahdabi et al. [24], a patent query is constructed by selecting the most representative terms from each section based on both log-likelihood weighting model and parsimonious language model [9]. While the authors only considered four sections, including title, abstract, description and claims, they draw the same conclusion that extracting terms from the description section of a patent document is the best way to generate queries. Mahdabi et al. [21] further proposed to utilize the international patent code as an additional indicator to facilitate automatic query generation from the description section of patents. Bouadjenek et al. [4] and Foletan et al. [8] used different approaches tried to remove the irrelevant terms from queries. However, the aforementioned approaches need to assign different weights to terms from different sections. In most cases, the weights of terms are difficult to obtain, and hence, they are assigned heuristically.

2.2.2 Patent query expansion

Patent search, as a recall-orientated search task, does not allow missing relevant patent documents. It is important to retrieve all possible relevant documents rather than finding only a small subset of relevant patents from the top ranked results. In order to improve the keyword coverage, a common practice is to enrich the query keywords, often referred to as *query expansion*. Recently, many query expansion techniques have been introduced in the field of patent search to improve the effectiveness of the retrieval. As discussed in Magdy and Jones [19], Manning et al. [25], Magdy [18], the methods for tackling this problem can be categorized into two major groups: (1) *similarity-based methods*, which either introduce similar terms or synonyms from patent documents or external resources, or extract new terms from patent documents to expand or reformulate queries, and (2) *feedback-based methods*, which modify the query based on the retrieved results, e.g., using pseudorelevance feedback or citation analysis.

Similarity-based methods try to append additional terms to the original keyword set. In practice, the additional terms can be extracted from either the query document or the external resources, e.g., WordNet and Wikipedia. For instance, Magdy and Jones [18,19] built a keyword-based synonym set with extracted synonyms and hyponyms from WordNet and utilized this synonym set to improve the retrieval performance. Mahdabi et al. [23] used definitions of the International Patent Classification to build a query-specific patent lexicon.

However, in some cases similarity-based methods cannot obtain reasonable results due to the deficiency of contextual information. The core of feedback-based methods is to employ user feedbacks to improve the quality of search results during the process of information retrieval. However, in practice, it is often difficult to obtain direct user feedbacks on the relevance of the retrieved documents, especially in patent retrieval. Hence, researchers usually exploit indirect evidence rather than explicit feedback of the search result. For example, pseudorelevance feedback (Pseudo-RF) [38], also known as blind relevance feedback, is a standard information retrieval technique that regards the top k ranked documents from an initial retrieval as relevant documents. It automates the manual process of relevance feedback so that the user gets improved retrieval performance without an extended interaction [25]. Although several related approaches have been proposed to employ pseudo-RF to facilitate the retrieval performance of patent search [13], existing studies indicate that those approaches perform relatively poor on patent retrieval tasks, as they suffer from the problem of topic drift due to the ambiguity and synonymity of terms [20].

2.2.3 Search results refinement

The rich content of a patent document consists of descriptions, embodiments, claims. The lexical content, as well as the structure of a patent document, is often the obstacle that makes patent documents difficult to read. To ease the understanding of patent documents, Shinmori et al. [30] utilized nature language processing methods to reduce the structural complexity. Sheremetyeva [29] proposed a similar approach to capture both the structure and lexical content of claims from US patent documents. Although they achieve a promising performance for improving the readability of patent documents, human efforts on comparing given patent documents are not significantly reduced.

Another research direction on refining search results is to use summarization techniques to represent original patent documents. Wang et al. [35] proposed an approach for generating comparative summarization via discriminative sentence selection. Wang and Li [34] utilized incremental hierarchy clustering for updating document summarization. Tseng et al. [33] utilized an extractive summarization method that selects sentences based on occurrence of keywords, title words and clue words contained in the document. Trappey et al. [32] employed a clustering-based approach that combines the ontological concepts and vector space model. The ontology captures the general concepts of patents in a given domain. Then, the proposed methodology extracts, clusters and integrates the content of a patent document to derive a summary and a tree diagram of key terms. These approaches might be able to capture the major information of a patent; however, they are not suitable to highlight the differences in two patent documents.

Our work is orthogonal to the aforementioned approaches, as we focus on the problem of comparing two patents. By presenting the comparative information, we are able to provide strong evidence for patent analysts of the difference between patent documents. Based on the provided evidence, patent analysts can quickly determine whether the idea of a patent application has been disclosed by previously granted patents, or whether a product-related patent document uses almost the same idea of another patent.

3 The framework of `PatSearch`

We present a framework, named `PatSearch`, to assist patent analysts in analyzing the patentability of patent documents. The proposed framework automatically transforms a patent

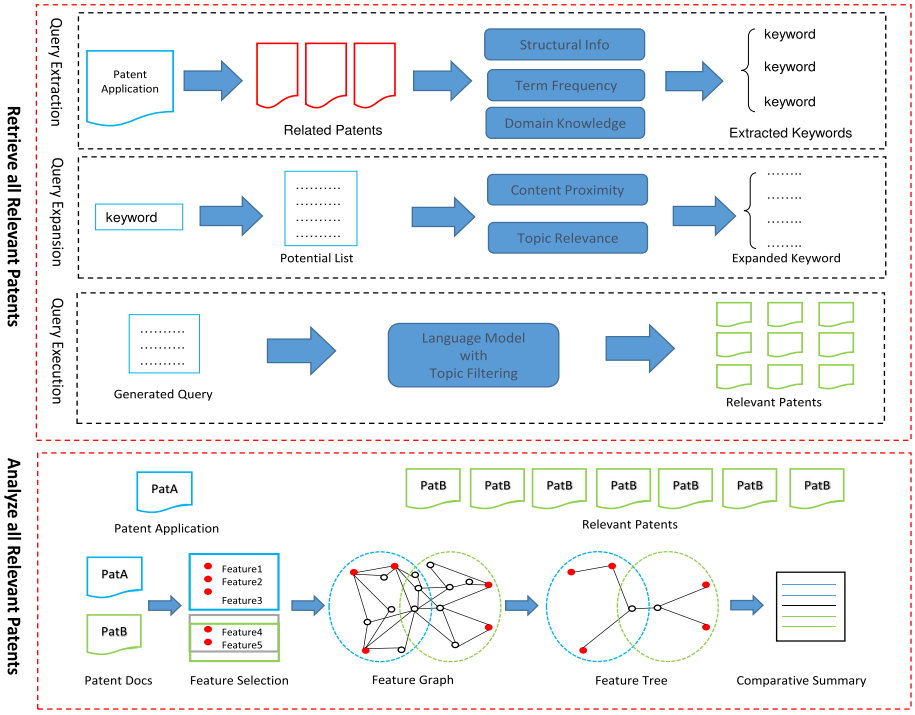


Fig. 2 System architecture of `PatSearch`

application into a reasonable and effective search query to provide enough information for patent analysts on making a patentability decision quickly. Figure 2 shows an overview of the framework of `PatSearch`. Specifically, it contains two major components: patentability search and patentability analysis.

3.1 Patentability search

In this module, we use the entire patent application as a query to retrieve all possible relevant patent documents. We first integrate multiple related information pieces from patents, such as classification code, patent content and structural information, to extract representative yet distinguishable terms from a given patent application. After that, in order to improve the retrieval performance, we expand the extracted terms by including additional words based on the content proximity and topic relevance. Finally, a list of relevant patent documents will be retrieved to assist patent analysts in making the patentability decision. Detailed explanation is provided in Sect. 4.

3.2 Patentability analysis

Patent documents are often lengthy, and full of technical and legal terminologies. Even for domain experts, it may also require a huge amount of time to read and analyze a single patent document. In this module, we present a new comparative summarization approach, which utilizes graph-based techniques to connect the dots among various aspects of the two patent

documents on a term co-occurrence graph, to help patent analysts quickly go through the retrieved results. Detailed explanation is provided in Sect. 5.

4 Retrieving relevant patents

In order to determine the patentability of a filing patent application, patent analysts often start with generating search queries. Traditionally, such search queries are generated by manually extracting keywords from the patent application first, followed by including additional technological terms. However, in many cases, this procedure requires a tremendous amount of time and human efforts, even for the most experienced domain experts. To address this challenge, we propose an approach to automatically transform the patent applications into search queries by finding all relevant patent documents for patentability analysis. The approach includes three modules: query extraction (Sect. 4.1), query expansion (Sect. 4.2) and query execution (Sect. 4.3).

4.1 Query extraction

In the framework, we extract important yet distinguishable keywords from a given patent application automatically by considering various aspects of the patent, such as the patent content, the classification code, as well as the structural information. We evaluate the quality of a term in the patent application using:

$$\frac{1}{n_f}\sum_{f=1}^{n_f} f(w, q_f) \cdot \log\left(1 + \frac{1}{f(w, D)}\right), \tag{1}$$

where f represents fields in patent documents, which are {title, abstract, description, claim}. $f(w, q_f)$ is the frequency of term w in the field f of patent application q, and $f(w, D)$ is the frequency of term w in the relevant patent document collection D (i.e., the patent documents that have at least one International Patent Classification code in common with the given patent application). The intuition behind Eq. (1) is that a term t, having a high average term frequency in all fields of the given patent application p, is more likely to be relevant to queries containing this term. Moreover, infrequent terms in the relevant patent documents have good discriminative capability, making them a better choice for describing the information content.

4.2 Query expansion

As mentioned above, to ensure the patentability of patent documents and maximize the scope of their protection, patent inventors and attorneys tend to use infrequent or domain-specific keywords to describe the corresponding inventions. Only considering the extracted keywords from patent applications is not sufficient to retrieve all relevant patent documents. To this end, in our framework, we propose a new approach, by combining the content proximity with topic relevance, to expand a given query.

4.2.1 Analyzing terms

Patent documents are legal documents and thus often have complex structures and technical contents. For example, in the claim part, a patent inventor may use “handoff” to describe

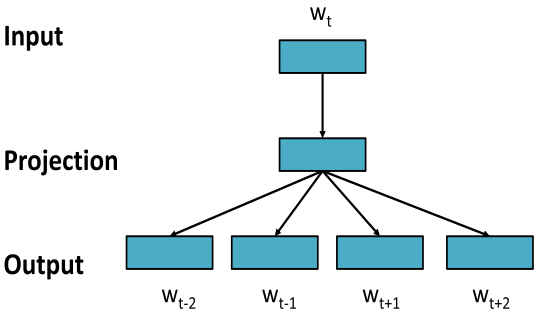


Fig. 3 Neural network architecture of the skip-gram model

the process of transferring an ongoing data session from one channel to another; however, in the description part, she may use "handover" to refer to the same technique. In order to help users quickly appreciate the technical concepts of a patent document and consequently improve the efficiency of patent retrieval, it is imperative to create a domain-related thesaurus by analyzing the corresponding technical terms.

In `PatSearch`, instead of using the bag-of-word representation model, we employ the skip-gram model [26], a new word-embedding approach for learning high-quality vector representations of words from a large amount of data. By using these vector representations of words, the keyword thesaurus is built to find the proximal terms to a given term, e.g., vec(handoff)$\approx$ vec(handover).

The skip-gram model generates the vector representation of words based on a language model obtained by building a neural network, avoiding the involvement of dense matrix multiplications. Figure 3 shows the neural network architecture of the skip-gram model, which consists of an input layer, a projection layer and an output layer to predict nearby words. Given a word sequence that contains word $w_1, w_2, \ldots, w_T$, the learning objective is to maximize the average log probability in a corpus, as shown in the following equation.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c}\log p(w_{t+j} \mid w_t), \tag{2}$$

where c is the size of the context window (in the experiment, we set window size $c = 2$) and $p(w_{t+j}|w_t)$ is defined by using the hierarchical softmax. The training speed of the skip-gram model could be billions of words per hour using modern computer because of the simple architecture and the use of negative sampling. We refer the interested readers to a nice tutorial on the skip-gram model.[3]

4.2.2 Analyzing topics

In some cases, a keyword might represent multiple meanings. For example, a "chip" may represent a "computer chip" or a "potato chip," and the corresponding patent documents with respect to these two meanings might be totally irrelevant. If patent document retrieval purely bases on keyword search, the results might not be reasonable due to the ambiguity of the keywords. We resolve this issue by discovering the underlining topics that occur in the document collection and perform patent retrieval based on the derived topics. A topic model is a type of statistical model for discovering the abstract "topics" in a collection

[3] http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/.

of documents. Specifically in PatSearch, we employ a well-known topic model, Latent Dirichlet allocation (LDA [3]) model, to extract topics from patent documents. LDA is a generative statistical model which assumes that each document is a mixture of a small number of topics and each word's creation is attributable to one of the document's topics.

Formally, we treat each patent abstract as a document d and assume that the generation of d is affected by a topic factor z, i.e., d is considered as a mixture of topics in the patent domain. Each topic corresponds to a multinomial distribution over the vocabulary. The existence of the observed word w in d is considered to be drawn from the word distribution specific to topic z, i.e., $p(w|z)$. Similarly, a topic z is drawn from the document-specific topic distribution d, i.e., $p(z|d)$. Based on the learned posterior probabilities, we are able to group the words contained in each patent abstract into semantic topics and therefore treat these topics as a knowledge base for further usage.

4.2.3 Query expansion

In order to improve the retrieval performance, we proposed a query expansion approach based on content proximity and topic relevance. The query expansion module first finds the top-k proximal terms (with $k = 10$) from term thesaurus for each term t in the given query q, to generate a potential expansion list L. Then, it alleviates the problem of topic drift by employing the topic-based approach to evaluate the topic relevance for each term l in the potential expansion list L with respect to the term t in the given query q. The relevance score (RS) between the query term t and the term l in the expansion list is calculated as follows.

$$RS(l, t) = \delta sim_{term}(l, t) + (1 - \delta) sim_{topic}(l, t), \tag{3}$$

where $sim_{term}(t, q)$ is the cosine similarity function based on word-embedding feature vectors $v_{term}(t)$ and $v_{term}(q)$, and $sim_{topic}(t, q)$ is the cosine similarity function based on word topic vectors $v_{topic}(t)$ and $v_{topic}(q)$. $\delta \in [0, 1]$ controls the relative importance of these two terms.[4]

4.3 Query execution

In the query execution module, given an expanded search query, PatSearch is able to retrieve all potential relevant patent documents and then filter them using the corresponding topics. We employ the latent topics to smooth the language models [37], and compute the similarity score of given query q for document d as follows:

$$score(q, d) = \lambda score_{topic}(q, d) + (1 - \lambda) score_{term}(q, d). \tag{4}$$

Equation (4) is a linear combination of the topic similarity and the term similarity, where the first term in Eq. (4) evaluates the similarity between query q and document d based on topic model, and the second term estimates the similarity in terms of the language model. $\lambda \in [0, 1]$ controls the relative importance of these two terms.[5] The first term in Eq. (4) is calculated as follows:

$$score_{topic}(q, d) = \prod_{\text{t in q}} \sum_{z=1}^{N} p(t|z) p(z|d).$$

[4] In the experiment, we empirically set δ as 0.5.

[5] In the experiment, we set λ to 0.3 as suggested in [14].

Here $p(z|d)$ and $p(t|z)$ are the posterior probabilities explained in Sect. 4.2.2. For the second term in Eq. (4), the language model, we employ the Dirichlet smoothed language model as follows:

$$score_{term}(q, d) = \prod_{w \in q} \frac{N}{N + 500} P(w|d) + \left(1 - \frac{N}{N + 500}\right) P(w|c),$$

where N is the number of tokens in document d, $P(w|d)$ is the maximum likelihood estimation of word w in document d and $P(w|c)$ is the maximum likelihood estimation of word w in the collection c.

5 Patent comparison

A major process in typical patent retrieval tasks is examining how similar/different two patent documents are from multiple aspects. It would be helpful if a comparative summary of the two patent documents being examined could be provided to ease the process. To this end, we model the problem of comparing patent documents as a summarization problem in Sect. 5.1 and propose a principled approach, called `PatentCom`, in Sect. 5.2. `PatentCom` can generate summaries to highlight both the commonalities and the differences in two patent documents.

5.1 Problem formulation

Suppose there are two patents $\mathbf{d}^1$ and $\mathbf{d}^2$ for comparison and each patent document is composed of a set of sentences, i.e., $\mathbf{d}^1 = \{s_1^1, s_2^1, \ldots, s_m^1\}$ and $\mathbf{d}^2 = \{s_1^2, s_2^2, \ldots, s_n^2\}$. The problem of comparing two patent documents is essentially a comparative summarization problem, i.e., to accurately discriminate the two documents by selecting a subset of sentences $\mathbf{s}^1 \subset \mathbf{d}^1$ and $\mathbf{s}^2 \subset \mathbf{d}^2$ with an identical summary length L. The generated comparative summaries $\mathbf{s}^1$ and $\mathbf{s}^2$ represent the general comparison of the major topic in $\mathbf{d}^1$ and $\mathbf{d}^2$, respectively. They can also be decomposed into several sections, each of which focuses on a specific aspect. For analysis purpose, the summaries should have both acceptable quality and wide coverage. In other words, the summaries should be representative and less redundant.

In general, a comparison identifies the commonalities or differences among objects. Therefore, a comparative summary should convey representative information in both documents and contain as many comparative evidences as possible. Specifically, given two documents, the comparative summarization problem is to generate a short summary for each document by extracting the most discriminative sentences, to deliver the differences between these documents. This problem is related to the traditional document summarization problem as both of them tend to extract sentences from documents to form a summary. However, traditional document summarization aims to cover the majority of information among documents, whereas comparative summarization is to discover the differences.

5.2 `PatentCom`: patent documents comparative summarization

In this paper, we propose a principled approach, named `PatentCom`, which utilizes graph-based methods to tackle the comparative summarization problem. Figure 4 presents an overview of our proposed approach. It contains four major modules, described as follows.

1. *Selecting discriminative features* (Sect. 5.2.1): Given two patents, we treat each document as a class and perform feature selection to extract discriminative terms (i.e., nouns).

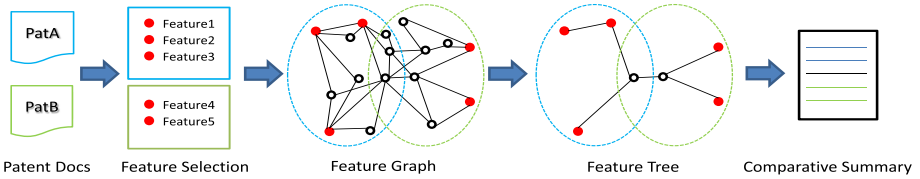


Fig. 4 An overview of `PatentCom`

2. *Constructing feature graph* (Sect. 5.2.2): We construct an undirected feature graph using the co-occurrence information of features in the original patent documents and map the discriminative features onto the graph.
3. *Extracting representative tree* (Sect. 5.2.3): Based on the discriminative features, we extract common information of two patents on the feature graph. The discriminative and common features are represented as a tree-based structure.
4. *Generating comparative summaries* (Sect. 5.2.4): We select sentences from the two patent documents by using the connected vertices on the generated feature tree. The resulted summary covers both commonalities and differences in patents.

5.2.1 Discriminative feature selection

Patent documents often differ from each other on specific aspects. For instance, technical patents often utilize different techniques in their inventions. Hence, we try to extract discriminative terms, i.e., nouns, from patent documents as the first step. These terms can be regarded as aspects that distinguish the two patents being compared. We therefore treat each patent document as a class, and nouns/noun phrases as features, resulting in modeling the problem as a feature selection problem.

Formally, suppose we have t feature variables from the two patent documents, denoted by $\{x_i | x_i \in F\}$, where F is the full feature index set, having $|F| = t$. The class variables are denoted as $C = \{c_1, c_2\}$. The problem of feature selection is to select a subset of features, $S \subset F$, based on which the target class variable C can be accurately predicted. There are various strategies to perform feature selection, e.g., information theory-based methods (such as information gain and mutual information) and statistical methods (such as χ^2 statistics). In our work, we adopt χ^2 statistics as the feature selection method as it has been successfully applied to the field of text mining [40].

5.2.2 Feature graph construction

The discriminative features from Sect. 5.2.1 are able to describe the differences between patents. However, a comparative summary of two patent documents should include both different and common aspects. To obtain the common aspects and link them to the differences, we resort to graph-based approaches.

Particularly in our work, we construct an undirected graph $\mathbb{G}$ to represent two patent documents, where $\mathbb{G} = (V, E)$. $\mathbb{G}$ contains a set of vertices (i.e., features) V, where each vertex represents the nouns/noun phrases in patent documents. Two vertices connect to each other only if they co-occur in the same sentence. In order to link two vertices, we consider both their co-occurrence and their corresponding frequencies in each document. Specifically, we define a linkage score of two vertices v_1 and v_2 in a single document A as

$$w_A(v_1, v_2) = 2 \times \frac{|\{(v_1, v_2)|v_1 \in A, v_2 \in A\}|}{|\{v_1|v_1 \in A\}| \times |\{v_2|v_2 \in A\}|}, \quad (5)$$

where $|\{v_1|v_1 \in A\}|$ and $|\{v_2|v_2 \in A\}|$ denote the frequencies of v_1 and v_2 in document A, respectively. $|\{(v_1, v_2)|v_1 \in A, v_2 \in A\}|$ represents the number of times that v_1 and v_2 appear in the same sentence of A. $w_A(v_1, v_2)$ essentially models the co-occurring probability of v_1 and v_2 in A. Given two patent documents A and B, v_1 and v_2 are connected if their average linkage score on both A and B exceeds a predefined threshold τ.[6]

5.2.3 Feature tree extraction

The discriminative features obtained from feature selection are capable of representing the difference in patent documents. However, there might be some gaps among these features, that is, they may not be well connected in the feature graph. In order to provide a fluent structure of comparative summary, we have to discover the relationship among discriminative features. This could be achieved by connecting the discriminative vertices and the vertices shared by two patent documents. Also, for presentation purpose, the generated summary should be as dense and informative as possible, i.e., to include the minimum number of features and convey the major commonalities/differences.

To address this problem, we formulate it as the minimum Steiner tree problem. The Steiner tree of some subset of the vertices of a graph $\mathbb{G}$ is a minimum-weighted connected subgraph of $\mathbb{G}$ that includes all vertices in the subset. Given a graph $\mathbb{G}$ (the feature graph in Sect. 5.2.2) and a subset of vertices S (the discriminative features in Sect. 5.2.1), the feature tree extraction is to find the Steiner tree of $\mathbb{G}$ that contains S with the minimum number of edges.

Given a graph $\mathbb{G} = (V, E)$, a vertex set $S \subset V$ (terminals) and a vertex $v_0 \in S$ from which every vertex of S is reachable in $\mathbb{G}$, the problem of minimum Steiner tree (MST) is to find the subtree of $\mathbb{G}$ rooted at v_0 that subsumes S with minimum number of edges.

The problem of MST is known as an NP-hard problem [12]. As suggested by [5], a reasonable approximation can be achieved by finding the shortest path from the root to each terminal and then combining the paths, with the approximation ratio of $O(\log^2 k)$, where k is the number of terminals. The approximation algorithm is described in Algorithm 1.

The algorithm employs a recursive way to generate the Steiner tree T. It takes a level parameter $i \geq 1$. When $i = 1$, $Steiner_1$ is simple to describe, i.e., to find the k closest terminals to the root v_0 and connect them to v_0 using shortest paths. As $i > 1$, $Steiner_i$ repeatedly finds a vertex v adjacent to the input root of the i-th function and a number k' such that the cost of the updated tree is the least among all trees of this form. Here the cost of a tree is calculated as the number of edges in the tree. After obtaining the expected path, we update the corresponding Steiner tree, the target size k and the terminal set S.

The generated Steiner tree of the feature graph gives us an elegant representation of patent comparison, which describes the transitions among all the other discriminative features, connected by the common features shared by two patents. Once the Steiner tree is generated, we can easily obtain a concise feature-based comparative summary of given patent documents.

5.2.4 Comparative summarization generation

The Steiner tree obtained from Sect. 5.2.3 provides us the basis to generate comparative summaries of two patent documents. Our goal is to select the minimum set of sentences from the original documents, by which the features in the Steiner tree can be fully covered.

[6] In the experiment, we empirically set τ as 0.1.

Algorithm 1 *Steiner*$_i(\mathbb{G}, S, v_0, k)$

Require: $\mathbb{G} = (V, E)$: an undirected features graph; S: terminal set; $v_0 \in S$: the root of the Steiner tree; k: the target size of terminals to be covered
Ensure: **T**: a Steiner tree rooted at r_0 covering at least k terminals
1: $T \leftarrow \emptyset$
2: **while** $k > 0$ **do**
3: $T_{opt} \leftarrow \emptyset$;
4: $cost(T_{opt}) \leftarrow \infty$
5: **for** v, $(v_0, v) \in E_{ct}$, and k', $1 \leq k' \leq k$ **do**
6: $T' \leftarrow Steiner_{i-1}(\mathbb{G}, S, v, k') \cup \{(v_0, v)\}$
7: **if** $(cost(T_{opt}) > cost(T'))$ **then**
8: $T_{opt} \leftarrow T'$
9: **end if**
10: **end for**
11: $T \leftarrow T \cup T_{opt}$; $k \leftarrow k - |S \cap V(T_{opt})|$;
12: $S \leftarrow S \setminus V(T_{opt})$
13: **end while**
14: **return** T

Each sentence can be represented as a subgraph of the entire feature graph, whereas the Steiner tree can also be regarded as a subgraph. Hence, the problem is to select the minimum set of subgraphs that cover the Steiner tree. Formally, we define the union of two graphs $G_a = (V_a, E_a)$ and $G_b = (V_b, E_b)$ as the union of their vertex and edge sets, i.e., $G_a \cup G_b = (V_a \cup V_b, E_a \cup E_b)$. We denote each sentence as $G_i = (V_i, E_i)$, which is a subgraph of $\mathbb{G}(V, \mathbf{w}_v, E, \mathbf{w}_e)$. We then formulate the problem of generating comparative summaries as the problem of finding the smallest subset of subgraphs whose union covers the Steiner tree. Given a graph $\mathbb{G} = (V, E)$, a set of subgraphs S and a Steiner tree T of $\mathbb{G}$, the subgraph cover problem (SGCP) is to find a minimum subgraph set $C \subset S$, whose union, $\mathbb{U} = (V_U, E_U)$, covers all the vertices and edges in T.

The SGCP problem is closely related to the set cover problem. The set cover problem (SCP), which is known as an NP-hard problem [12], can be easily reduced to the SGCP problem.

REDUCTION. Given a universe U, a set of elements $\{1, 2, \ldots, m\}$ and a family S of subset of U, we generate a fully connected graph $\mathbb{G} = (V, E)$ for each subset, where nodes are elements of subset and every pair of nodes has an edge. This construction can be done in polynomial time in the size of set cover instance.

Assume the universe U has a cover C with length k, where C is a smallest subfamily $C \subset S$ of sets whose union is U. Based on set cover C, we generate a set S of a fully connected graph G_i, where the vertex set of G_i is the same with C_i. Suppose we have a graph $T = (V_T, E_T)$, the vertex set V_T equals to the union of C. It is straightforward that the set S is the cover of T, because T is a subgraph of union of S and there is no smaller set of subgraph to cover all the vertices in T.

For the reverse direction, assume that $T = (V_T, E_T)$ has a subgraph cover S with length k. Let us only consider the vertex part of S, and we can get a set C of k sets whose union equals V_T, the universe. This set will cover the universe, and thus, the subgraph cover in $\mathbb{G}$ is a set cover in U. □

The greedy algorithm for the set cover problem chooses sets according to the following rule: Choose the set that contains the largest number of uncovered elements at each iteration. It has been shown [6] that this algorithm gets an approximation ratio of $H(s)$, where s is the size of the set to be covered and $H(m)$ is the m-th harmonic number:

$$H(m) = \sum_{j=1}^{m} \frac{1}{j} \leq \ln m + 1$$

6 Experimental evaluation

In this section, we provide a comprehensive experimental evaluation to demonstrate the efficacy and effectiveness of our proposed framework `PatSearch`. We start with an introduction to the patent collection used in the experiment. To evaluate our proposed framework, we compare our method with other existing solutions. Finally, we conduct a case study on the patent application regrading "optical panel" to demonstrate the idea of patent comparison.

6.1 Data collection

For relevant patent search, there is no standard benchmark data set for completed patent search tasks, which can provide the ground truth of relevant patent documents with respect to a patent application. The real data set used in our experiments is obtained from the United States Patent and Trademark Office,[7] including 1,847,225 US granted patents, whose filing dates range from 2001 to 2012. Similar to the strategy used in the NTCIR workshop series [31], we consider the citation field of these patents as a substitute in terms of relevance judgments for evaluation purpose. These references are usually assigned by examiners during patent prosecution. But it is quite common in practice that truly relevant patents are not cited. Although the strategy of using citations as relevance judgments has a number of limitations, the same setting affects all patent retrieval algorithms. Therefore, it provides a reasonable basis for comparing and evaluating algorithms in patent retrieval. We discard the citations to non-US patents and non-patent literature and also do not include references to US patents that are not covered in data collection.

We extract the four fields, i.e., title, abstract, claims, and description, and preprocess these contents using natural language processing techniques including stopword removal, tokenizing and stemming. The number of tokens is more than 14 billions, and the size of vocabulary is more than 8 millions. The Lucene[8] toolkit is employed for text indexing. DL4J[9] library is used to build the keywords repository, where the number of vector is fixed to be 1000. The Mallet[10] library is employed to build the topic model among the patent collection, where the number of topics is set to be 1000. The test query set for patent retrieval is built by randomly selecting 100 patents that have at least 20 citations.

Patent document comparison is a relatively new application in patent retrieval, and there is still no benchmark data set for the evaluation. Note that patent comparison is usually done by experienced patent attorneys and it often needs a large amount of billing hours. In this paper, we are provided a patent comparative summarization data set by a patent agent company and the data set is generated according to the real-world patentability or infringement analysis reports from the company. The data set is composed of 300 pairs of US patents related to various topics, including "DOMESTIC PLUMBING," "OPTICS DEVICE OR ARRANGEMENT," "INFORMATION STORAGE." For each comparable patent pair, manual summaries are provided by three patent attorneys as the references.

[7] http://www.uspto.gov/.

[8] http://lucene.apache.org/.

[9] http://deeplearning4j.org/.

[10] http://mallet.cs.umass.edu/.

6.2 Evaluation methodology

To evaluate our proposed framework, we implement two existing methods for query expansion:

- *WordNet* [19]: It employs WordNet to extract the synonyms and hyponyms for each term in the search query. WordNet is a large lexical database of English that groups different terms into sets of cognitive synonyms. It is often employed by researchers from the information retrieval community to enhance retrieval effectiveness.
- *Pseudorelevance feedback(PRF)* [38]: Pseudorelevance feedback, also known as blind relevance feedback, is a standard retrieval technique that regards the top k ranked documents from an initial retrieval as relevant documents. After an initial run of a given query q_0, it uses the Rocchio [27] algorithm to generate a modified query q_m.

$$q_m = \alpha q_0 + \beta \frac{1}{|D_r|} \sum_{d_j \in D_r} d_j - \gamma \frac{1}{|D_{nr}|} \sum_{d_j \in D_{nr}} d_j$$

where q_0 defines the original query vector and D_r and D_{nr} are the set of relevant and irrelevant documents, respectively. We set the weights variable $\alpha = 1$, $\beta = 0.75$, $\gamma = 0.15$ and consider the top-20 retrieved documents as relevant documents and others as irrelevant documents.

We also implement three document summarization methods, including

- *Minimal dominate set model (MDSM)* [28], which selects the most representative sentences from each patent document;
- *Discriminative sentence selection model (DSSM)* [36], in which the selection is modeled as an optimization problem that minimizes the conditional entropy of the sentence membership given the selected sentence set.
- *Comparative summarization via linear programming model (CSLPM)* [10], which considers cross-topic concept pairs as comparative evidences and topic-related concepts as representative evidences. Then, the quality of a comparative summary is evaluated using

$$\lambda \sum_{j=1}^{|C_1|} \sum_{k=1}^{|C_2|} u_{jk} \cdot op_{jk} + (1-\lambda) \sum_{i=1}^{2} \sum_{j=1}^{|C_i|} w_{ij} \cdot oc_{ij}, \tag{6}$$

which is a linear combination of the representativeness and the comparative importance.

Joho [11] conducted a survey on patent users to show that the patent examiners are willing to review the top 100 patents. So in this paper, we adopt recall, F1 score, and mean average precision (MAP) to evaluate the performance of patent retrieval on the top-100 retrieved relevant patents with baseline methods.

- *Recall* [1]: It is the ratio of the number of retrieved relevant patents to all the relevant patents.

$$Recall = \frac{|\,(relevantitemsretrieve)\,|}{|relevantitems|} \tag{7}$$

- *F1 score* [1]: It is a measure that trades off between precision and recall, which is the evenly weighted.

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{8}$$

– *Mean Average Precision (MAP)* [1]: It is the mean of average precision for all test patents.

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{m_j}\sum_{k=1}^{m_j} Precision(R_{jk}) \tag{9}$$

where R_{jk} is the set of ranked retrieval results from the top of retrieved list to item k in the list, and the set of relevant documents for query $q_1 \in Q$ is $/p_1, p_2,\ldots,p_{m_i}/$. If a relevant document is not occurred in retrieval list, the precision value is 0.

Furthermore, we use ROUGE [15] as the metric to evaluate the quality of the generated summaries, which has been widely used in document summarization evaluation. Given a system generated summary and a set of reference summaries, ROUGE measures the summary quality based on the unit overlap counting. In the experiment, for each summarization method, we calculate the averaged scores of ROUGE-1, ROUGE-2, ROUGE-W and ROUGE-SU over 300 pairs of patent documents.

6.3 Results and analysis

6.3.1 Query extraction performance

In `PatSearch`, we extract top-30 important terms to form an initial search query from the given patent documents based on our query extraction module. To evaluate our query extraction approach, we compare the generated initial search query with some baseline methods to demonstrate the effectiveness. The baseline methods include using title (TIL), abstract (ABS), claim (CLM) and the entire patent document (ALL) as the search query. The results are reported in Table 1.

Symbol * in Table 1 indicates the statistical significant improvement over the baselines in terms of recall, MAP and F1 score. As previously explained in Sect. 4.1, our search query generation method selects terms which either have a high average term frequency in all fields or have more discriminative ability due to the infrequency in the relevant patent documents. As depicted in Table 1, our generated search query achieves the best performance compared with other baseline methods in terms of recall, MAP and F1 score. Especially for the recall, it significantly outperforms other methods. A higher recall is valuable, because it means less human efforts and lower risk of missing important patent documents. The reason is apparent: Applicants often describe the abstract/claims without enough technical details in order to decrease the retrievability of their patents, not to mention the fact that the terms in the abstract/claims often contain obscure meanings.

Table 1 Performance for query generation

	Recall	MAP	F1 score
TIL	0.153	0.044	0.054
ABS	0.185	0.052	0.066
CLM	0.169	0.047	0.06
ALL	0.215	0.058	0.077
PatSearch without expansion	0.254*	0.06*	0.09*

Table 2 Performance for query expansion

	Recall	MAP	F1 score
WordNet	0.185	0.063	0.104
PRF	0.169	0.058	0.088
PatSearch without expansion	0.254	0.06	0.09
PatSearch	0.385*	0.082*	0.137*

6.3.2 Query expansion performance

In patent retrieval, it is important to retrieve all possible relevant documents rather than finding only a small subset of relevant patents from the top ranked results. In `PatSearch`, given the generated search query, our query expansion module selects top-3 relevant terms for expansion based on the combination of content proximity and topic relevance. For comparison, we compute the recall, F1 score and MAP with baseline methods for query expansion including ones using WordNet and pseudorelevance feedback. The results are reported in Table 2. Symbol * denotes statistical significant improvements.

The main observation from Table 2 is that our query expansion approach is always more effective than the other three methods. In addition, our approach improves the baseline in terms of recall significantly. Based on the analysis, we observe that the query expansion within WordNet slightly improves the retrieval performance. However, in some cases, it cannot obtain satisfied results due to the deficiency of contextual information. The query expansion using pseudorelevance feedback (PRF) performs relatively poor on patent retrieval tasks, as it suffers from the problem of topic drifting, i.e., the topics of the query may change/shift to an unintended direction after query expansion, due to the ambiguity and synonymity of terms.

6.3.3 Patent comparison performance

A typical patent document often contains multiple sections, including summary of the invention, description of the preferred embodiments, claims. Some sections may describe the invention in more details, whereas others may represent the invention using abstractive terms. To evaluate the importance of each section in delivering the comparative information, we generate the comparative summaries from different sections of patent documents, e.g., claims (CLM), embodiments (EMB), summary of the invention (SUM), the combinations of these three sections and the entire patent document (ALL).

In Table 3, we report the averaged ROUGE scores of `PatentCom` for the summaries generated from different sections of patent pairs. We observe that the best score is achieved by the summaries generated from combination of embodiment section and claims. The reason is that the claim section is the core part of the entire patent document, while the embodiment of a patent document describes in detail how the invention can be implemented and practiced, which contains sufficient resources to generate a comparative summary. Besides, it is not appropriate to consider them separately, because the claim part is generally full of legal or domain-specific terminologies, while the embodiment part contains detailed information with less significance.

Table 4 shows the comparison results of different summarization methods, which are averaged ROUGE scores over 300 pairs of patent documents. We observe that (1) `PatentCom` achieves the best performance in terms of all the ROUGE scores by considering both com-

Table 3 Comparison of using different sections

Sections	ROUGE-1	ROUGE-2	ROUGE-W	ROUGE-SU
CLM	0.54243	0.3306	0.15522	0.22305
SUM	0.48311	0.26425	0.11398	0.19616
EMB	0.44775	0.23178	0.09728	0.146
CLM + SUM	0.59384	0.41746	0.20378	0.28874
CLM + EMB	0.60782	0.46232	0.22446	0.31129
EMB + SUM	0.49882	0.30076	0.12704	0.21713
ALL	0.60531	0.45934	0.22267	0.3093

Table 4 Comparison of different models

Models	ROUGE-1	ROUGE-2	ROUGE-W	ROUGE-SU
MDSM	0.52102	0.30999	0.14993	0.28867
DSSM	0.46049	0.26451	0.11484	0.15833
CSLPM	0.53092	0.40663	0.21189	0.30150
PatentCom	0.60531	0.45934	0.22267	0.30930

monalities and differences between two patent documents; (2) the performance of DSSM is not comparable with the other two methods, indicating that only considering the difference in the patent pair is not sufficient for this task, since such difference may not be significant or comparable; and (3) MDSM has similar ROUGE-1 scores with CSLPM, since MDSM selects important sentences for each patent so that the summaries generated by MDSM contain frequent words in patents and may have significant overlap with reference summaries based on unigram. However, MDSM performs poorly on ROUEG-2, ROUGE-W and ROUGE-SU scores, as its purpose is not consistent with the objective of this task.

6.4 An illustrative case study for determining patentability

We conduct a real-world case study of determining patentability on a patent application US2013,0301,299 (US299) to demonstrate the efficacy of our framework, `PatSearch`. Given the patent application US299, patent retrieval module firstly extracts the important terms such as "light," "fabricating," "photolithography." And then, for each extracted term, a expansion query list is provided to refine the initial search query, which is generated by our query expansion method. For example, the expand query for the term "light" are "photo," "radiate," "ultraviolet," "optical," etc. Moreover, a ranked list of relevant patent documents is retrieved to help patent analysts determine patentability. The comparative summaries are generated by patent comparison module for each retrieved result. Table 5 shows two comparative summaries of two retrieved patents US7,094,520 (as US520), which ranked as 11, and US6,663,253 (as US253), which ranked as 87, with respect to patent application US299.

In the comparative summary of US253 and US299, US253 mentions coating the mole base by the photosensitive heat-resistant resin, while US299 mentions forming a layer of photosensitive material on a mold, and the photosensitive material is photosensitive resist. As we can see, these two procedures are very similar. In the comparative summary of US520 and US299, US520 mentions the procedure of building photosensitive heat-resistant resin layer

Table 5 A sample comparative summary for patentability analysis

Patent	US253	US299
	The formation of the **molded pattern** on the **mold base** by the use of the positive-type **photosensitive heat-resistant resin** comprises the steps of coating the **mold base** with the positive-type photosensitive heat-resistant resin to form the photoresist film on its surface, preheating the photoresist film so as to harden slightly, exposing the applied photoresist film to light via the positive-type **pattern film** for forming the **optical pattern**	Claim 1. A fabricating method of grid points on a **light guiding plate**, comprising following steps of: S1, forming a layer of **photosensitive material** on a mold for the light guiding plate; and S2, performing **photolithography** on the photosensitive material in order to form **grid points** on the light guiding plate. Claim 2. The method according to claim 1, wherein the photosensitive material is a **photosensitive resist**
Patent	US520	US299
	A development step in which the **photosensitive heat-resistant resin layer** 12 exposed is developed; a rinsing step in which the portions removed by the development are rinsed away; and a baking step in which the pattern formed by the development is baked at a high temperature to cure the photosensitive heat-resistant resin and form a raised or depressed **pattern**...	Claim 5. The method according to claim 2, wherein the step of S2 further comprises following steps of: S21 using a film formed with **grid points** arrangement **pattern** as a mask, S22 sequentially performing exposing and developing process on the photosensitive resist in order to form a grid points pattern on the photosensitive resin, and S23 curing the photosensitive resist and removing **residual solvent** and moisture

The patent application being analyzed is US299, which can be partially covered by US253 and US520

and a rinsing step, while US299 mentions similar procedure of removing residual solvent. From the selected comparative summarizes, we observe that the combination of US520 and US253 discloses similar process for producing an optical panel molding die, which is described as light guild panel in US299. Such summaries provide informative information to patent analysts that there is a high probability that the claims 1, 2, 5 in patent application US299 might be rejected under pre-AIA 35 USC 103(a)(non-obvious) as being unpatentable over US520 in view of US253.

7 Conclusion and future work

In this paper, we study the problem of determining the patentability for a given patent application. Based on the analysis of domain characteristics of patents, we propose a unified framework, called `PatSearch`, to help patent analysts in making the patentability decision in a systematic way. The framework automatically extracts representative yet distinguishable terms to generate a search query for the given patent application. We further propose a new query expansion method to alleviate the issues of ambiguity and topic drifting. Finally, a comparative summarization technique is proposed to reduce human efforts of comparing patent documents. Extensive empirical evaluation and case studies on a collection of US patents demonstrate the efficacy and effectiveness of our proposed framework.

In our proposed framework, the representativeness of terms is defined based on high average term frequency in all fields of the given patent application and the weights of different fields are set equally. It is thus interesting to determine the weights using external resources, such as learning the weights from historical data collection, and domain knowledge. Further, to expand the generated query terms, the relevance scores based on the content proximity and topic relevance are calculated. An interesting direction is to consider the relationship between query terms and expanded terms to improve the coverage and effective of the patent search. Finally, in the domain of patentability search, there is no benchmark dataset of evaluating patentability search tasks. It is thus worthy to build such benchmark datasets to evaluate the state-of-the-art techniques in real-world patentability search tasks based on patentability analysis report from WIPO.

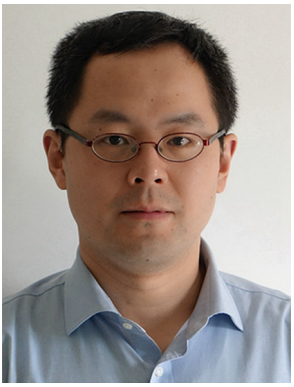
Acknowledgements We thank anonymous reviewers for their very useful comments and suggestions. The work was supported by National Science Foundation of China under Grant 91646116, Ministry of Education/China Mobile Joint Research Fund under Project 5-10, Jiangsu Provincial Natural Science Foundation of China under Grant BK20171447, Jiangsu Provincial University Natural Science Research of China under Grant 17KJB520024, Nanjing University of Posts and Telecommunications under Grant NY214135 and NY215045.

References

1. Alberts D, Yang CB, Fobare-DePonio D, Koubek K, Robins S, Rodgers M, Simmons E, DeMarco D (2017) Introduction to patent searching. In: Lupu M, Mayer K, Kando N, Trippe A (eds) Current challenges in patent information retrieval. The information retrieval series, vol 37. Springer, Berlin, Heidelberg
2. Atsushi H, Yukawa T (2004) Patent map generation using concept-based vector space model. Working notes of NTCIR-4, Tokyo, pp 2–4
3. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
4. Bouadjenek MR, Sanner S, Ferraro G (2015) A study of query reformulation for patent prior art search with partial patent applications. In: Proceedings of the 15th international conference on artificial intelligence and law. ACM, pp 23–32
5. Charikar M, Chekuri C, Goel A, Guha S (1998) Rounding via trees: deterministic approximation algorithms for group Steiner trees and k-median. In: Proceedings of the 30th annual ACM symposium on theory of computing. ACM, pp 114–123
6. Chvatal V (1979) A greedy heuristic for the set-covering problem. Math Oper Res 4(3):233–235
7. Fujii A (2007) Enhancing patent retrieval by citation analysis. In: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 793–794
8. Golestan Far M, Sanner S, Bouadjenek MR, Ferraro G, Hawking D (2015) On term selection techniques for patent prior art search. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 803–806
9. Hiemstra D, Robertson S, Zaragoza H (2004) Parsimonious language models for information retrieval. In: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 178–185
10. Huang X, Wan X, Xiao J (2011) Comparative news summarization using linear programming. ACL-HLT, ACL, pp 648–653
11. Joho H, Azzopardi LA, Vanderbauwhede W (2010) A survey of patent users: an analysis of tasks, behavior, search functionality and system requirements. In: Proceedings of the 3rd symposium on information interaction in context. ACM, pp 13–24
12. Karp RM (1972) Reducibility among combinatorial problems. Springer, Berlin
13. Kishida K (2003) Experiment on pseudo relevance feedback method using Taylor formula at NTCIR-3 patent retrieval task. In: Proceedings of the 3rd NTCIR workshop on research in information retrieval, automatic text summarization and question answering. NII, Tokyo. http://research.nii.ac.jp/ntcir
14. Krestel R, Smyth P (2013) Recommending patents based on latent topics. In: Proceedings of the 7th ACM conference on recommender systems. ACM, pp 395–398
15. Lin C-Y, Hovy E (2003) Automatic evaluation of summaries using n-gram co-occurrence statistics. NAACL-HLT, ACL, pp 71–78
16. Lupu M, Hanbury A et al (2013) Patent retrieval. Found Trends Inf Retr 7(1):1–97

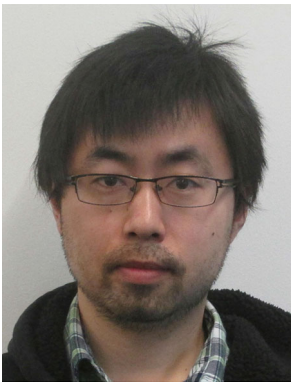
17. Lupu M, Mayer K, Tait J, Trippe AJ (2011) Current challenges in patent information retrieval. Springer Science & Business Media, Berlin
18. Magdy W (2012) Toward higher effectiveness for recall-oriented information retrieval: a patent retrieval case study. PhD thesis, Dublin City University
19. Magdy W, Jones G (2011) A study on query expansion methods for patent retrieval. In: Proceedings of the 4th workshop on patent information retrieval. ACM, pp 19–24
20. Magdy W, Leveling J, Jones GJF (2009) Exploring structured documents and query formulation techniques for patent retrieval. In: Peters C, Di Nunzio GM, Kurimo M, Mandl T, Mostefa D (eds) Proceedings of the 10th cross-language evaluation forum conference on Multilingual information access evaluation: text retrieval experiments (CLEF'09). Springer-Verlag, Berlin, Heidelberg, pp 410–417
21. Mahdabi P, Andersson L, Keikha M, Crestani F (2012) Automatic refinement of patent queries using concept importance predictors. In: Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 505–514
22. Mahdabi P, Crestani F (2014) Patent query formulation by synthesizing multiple sources of relevance evidence. ACM Trans Inf Syst (TOIS) 32(4):16
23. Mahdabi P, Gerani S, Huang JX, Crestani F (2013) Leveraging conceptual lexicon: query disambiguation using proximity information for patent retrieval. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 113–122
24. Rauber A, de Vries AP (eds) Multidisciplinary information retrieval. IRFC 2011. Lecture Notes in Computer Science, vol 6653. Springer, Berlin, Heidelberg
25. Manning C, Raghavan P, Schütze H (2008) Introduction to information retrieval, vol 1. Cambridge University Press, Cambridge
26. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ (eds) Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13), vol 2. Curran Associates Inc., USA, pp 3111–3119
27. Salton G (1971) The SMART retrieval system—experiments in automatic document processing. Prentice-Hall, Inc., Upper Saddle River, NJ, USA
28. Shen C, Li T (2010) Multi-document summarization via the minimum dominating set. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10). Association for Computational Linguistics, Stroudsburg, PA, USA, pp 984–992
29. Sheremetyeva S (2003) Natural language analysis of patent claims. In: Proceedings of the ACL-2003 workshop on patent corpus processing-volume 20. Association for Computational Linguistics, pp 66–73
30. Shinmori A, Okumura M, Marukawa Y, Iwayama M (2003) Patent claim processing for readability: structure analysis and term explanation. In: Proceedings of the ACL-2003 workshop on patent corpus processing-volume 20. Association for Computational Linguistics, pp 56–65
31. Takeuchi H, Uramoto N, Takeda K (2005) Experiments on patent retrieval at NTCIR-5 workshop. NTCIR-5
32. Trappey AJ, Trappey CV, Wu C-Y (2009) Automatic patent document summarization for collaborative knowledge systems and services. J Syst Sci Syst Eng 18(1):71–94
33. Tseng Y, Lin C, Lin Y (2007) Text mining techniques for patent analysis. Inf Process Manag 43(5):1216–1247
34. Wang D, Li T (2010) Document update summarization using incremental hierarchical clustering. In: Proceedings of the 19th ACM international conference on information and knowledge management, CIKM'10. ACM, New York, pp 279–288. https://doi.org/10.1145/1871437.1871476
35. Wang D, Zhu S, Li T, Gong Y (2012) Comparative document summarization via discriminative sentence selection. ACM Trans Knowl Discov Data 6(3):12:1–12:18. https://doi.org/10.1145/2362383.2362386
36. Wang D, Zhu S, Li T, Gong Y (2012) Comparative document summarization via discriminative sentence selection. ACM Trans Knowl Discov Data 6(3):12
37. Wei X, Croft WB (2006) LDA-based document models for ad-hoc retrieval. In: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 178–185
38. Xu J, Croft WB (1996) Query expansion using local and global document analysis. In: Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 4–11
39. Xue X, Croft W (2009) Transforming patents into prior-art queries. In: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval. ACM, pp 808–809
40. Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. ICML 97:412–420
41. Zhang L, Li L, Li T (2015) Patent mining: a survey. ACM SIGKDD Explor Newsl 16(2):1–19

Longhui Zhang received the Ph.D. degree in computer science from the School of Computing and Information Sciences, Florida International University, Miami, FL, in 2016. He is currently a co-founder and data scientist in the Deep Fusion Technologies LLC. His research area includes information retrieval, data mining and machine learning. He is also a qualified patent attorney at China and completed the study in the fields of the American legal system and intellectual property law from the John Marshall Law School.

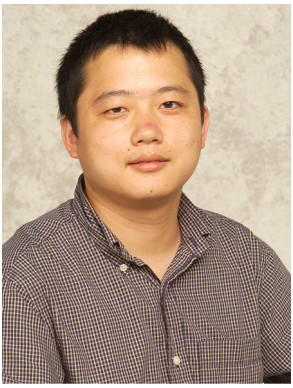


Zheng Liu received the Ph.D. degree from the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, in 2011. He is currently an assistant professor in the School of Computer Science, Nanjing University of Posts and Telecommunications, China. His research interests include summarizing and querying large graph data and mining large-scale network management data. He has published research papers in several major conferences and journals, including ICDE, ICDM, DASFAA, KIS.

Lei Li received the M.S. degree in software engineering from Beihang University, Beijing, China, in 2008, and the Ph.D. degree in Computer Science from Florida International University in 2014. His research interests include data mining, machine learning and recommender systems.



Chao Shen received B.S. and M.S. in Computer Science from Fudan University in 2006 and 2009 and Ph.D. in Computer Science from Florida International University in 2014. He has been working on natural language processing and data mining.



Tao Li received the Ph.D. degree in Computer Science from the Department of Computer Science, University of Rochester, Rochester, NY, in 2004. He is a professor with the School of Computing and Information Sciences, Florida International University, Miami, FL. His research interests include data mining, computing system management, information retrieval and machine learning. He received the US National Science Foundation (NSF) CAREER Award and multiple IBM Faculty Research Awards.