



Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Unsupervised technical phrase extraction by incorporating structure and position information

Peng Zhou, Xin Jiang, Shu Zhao*

Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Anhui Province, China
 School of Computer Science and Technology, Anhui University, Hefei, Anhui Province, China
 Information Materials and Intelligent Sensing Laboratory of Anhui Province, Anhui Province, China

ARTICLE INFO

Keywords:

Patent text mining
 Technical phrase extraction
 Graph construction
 Structure and position information

ABSTRACT

The vigorous development of patent applications in recent years provides an opportunity to unveil the inherent laws of innovation, but it also puts forward higher requirements for patent mining technology. An essential step for patent text mining is to establish a technology portrait for each patent, that is, identify the technical phrases involved, which can be summarized and represented by the patent from the technical point of view. Currently, there is a large body of work focusing on keyword extraction. However, technical phrase extraction differs from keyword extraction due to the unique properties of technical phrases. Specifically, technical phrases must contain rich technical information and are essential to the entire patent text from a technical perspective. Meanwhile, finding potential relationships between phrases with different technical meanings is challenging for technical phrase extraction. Based on the analysis of the characteristics of technical phrases, we found that the position of technical phrases in the patent text and the structural relationship between technical phrases are crucial, and how to make good use of these two pieces of information is a challenge. Motivated by this, we propose a new Unsupervised Technical phrase Extraction model from the Structure and Position information perspective, named UTESP. Specifically, UTESP includes four key steps: candidate generation, graph construction, candidate score, and candidate selection. The structure information refers to adjusting the incoming edge weight of candidate phrases through the distance relations between candidate phrases and applying the graph ranking algorithm to obtain the structure score of the candidate phrase. The position information simultaneously incorporates the position and frequency of candidate phrases in the patent text to calculate a position score for candidate technical phrases. The effectiveness of our framework has been demonstrated by comparing with seven competitive algorithms on the patent datasets in terms of three evaluation metrics: Precision, Recall, and F1 scores. Besides, our new framework indicated significant improvements in the representation ability of technical phrases by comparing Information Retrieval Efficiency (IRE) with competitive algorithms.

1. Introduction

Since 2004 (except 2009), WIPO (World Intellectual Property Organization) indicates that the number of patent applications in the world has increased every year (Liu, Hseuh et al., 2011). According to the State Intellectual Property Office statistics in China, from 2014 to 2019, the number of domestic patent applications increased explosively, from 868,500 in 2014 to 2.268 million in 2019, an increase of 2.6 times in five years (Zhang et al., 2014). The explosive growth has brought a valuable database for revealing the inherent laws of innovation, but it has also put forward higher requirements for patent mining technology (Wu et al., 2019).

Intellectual property consists of many parts, of which patents account for a large part. A useful patent mining can bring huge benefits to enterprises (Liu et al., 2018). Through patent mining, we can comprehensively and effectively protect achievements and peripherally related technologies that may have patent application value, and avoid loopholes in patent protection. Essential patents that threaten competitors can be discovered as early as possible, which makes it easier for companies to design avoidance to avoid patent risks (Zhang et al., 2015). From the perspective of the objects mined, it can be roughly divided into two parts: patent metadata mining and patent text mining. Compared with the latter, the former is more mature in mining methods and technologies (Zhang et al., 2015). However, emerging technical

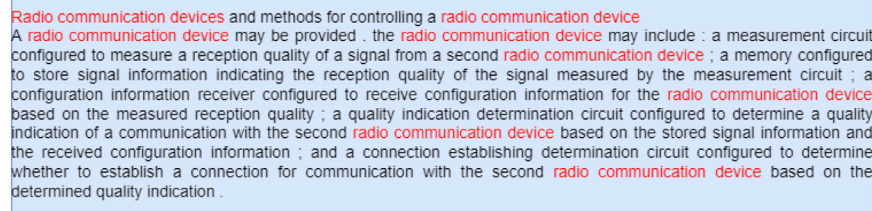
* Corresponding author at: School of Computer Science and Technology, Anhui University, Hefei, Anhui Province, China.
 E-mail addresses: doodzhou@ahu.edu.cn (P. Zhou), jiangxin@stu.ahu.edu.cn (X. Jiang), zhaoshuzs@ahu.edu.cn (S. Zhao).

<https://doi.org/10.1016/j.eswa.2024.123140>

Received 9 April 2023; Received in revised form 8 November 2023; Accepted 2 January 2024

Available online 5 January 2024

0957-4174/© 2024 Elsevier Ltd. All rights reserved.



Radio communication devices and methods for controlling a radio communication device
 A radio communication device may be provided : the radio communication device may include : a measurement circuit configured to measure a reception quality of a signal from a second radio communication device ; a memory configured to store signal information indicating the reception quality of the signal measured by the measurement circuit ; a configuration information receiver configured to receive configuration information for the radio communication device based on the measured reception quality ; a quality indication determination circuit configured to determine a quality indication of a communication with the second radio communication device based on the stored signal information and the received configuration information ; and a connection establishing determination circuit configured to determine whether to establish a connection for communication with the second radio communication device based on the determined quality indication .

Fig. 1. An example of technical phrases (in red color) in a patent text. Important technical phrases are always near the beginning of the patent text and appear multiple times. Meanwhile, technical phrases contain much technical information that differs from traditional keywords.

phrases often appear in patent texts (Hu et al., 2018). An effective step of patent text mining is to establish a technical portrait for each patent, in other words, to determine the technical phrases appearing in the patent text, and to summarize the technical information of the patent text from a technical perspective (Liu et al., 2020).

Many works are designed for professional phrase extraction (Papa-
 giannopoulou & Tsoumakas, 2020). Text mining technology can be divided into many types, mainly including the following four categories: keyphrase extraction, named entity recognition, concept extraction and technical phrase extraction. Specifically, keyphrase extraction wants to extract the key information that can summarize the text, and these phrases prefer those that frequently appear and are close to the topic, such as Li et al. (2021) and Yu and Ng (2018). The purpose of named entity recognition is to classify the extracted phrases into different categories (Akbik et al., 2019; Yu et al., 2020). Concept extraction is widely used in medical treatment, and its purpose is to find words or phrases that describe concepts (Fang et al., 2021; Yang, Bian et al., 2020). Technical phrase extraction aims to be able to describe and summarize patent texts (Liu et al., 2020). Compared with key phrase extraction, NER (Named Entity Recognition), and concept extraction, technical phrase extraction is more like a combination of key phrases and concepts but is different from them. Technical phrase extraction not only summarizes patents but also has specific technical meanings. The technical phrases are vital components of patent texts, which describes some key and technical information in specific fields (Hu et al., 2018). Compared with general literature, the characteristics of patent text are different. For example, emerging technical phrases always appear many times and appear relatively early in the patent text. Besides, the format of the patent text is more strict and detailed than general documents.

To our knowledge, only a few works are dedicated to technical phrase extraction. Liu et al. (2020) proposed an unsupervised technical phrase extraction framework and summarized some characteristics of technical phrases, such as part of speech, number of words, semantic context, and global occurrence. However, we found that it needed to be more comprehensive. Fig. 1 illustrates an example of technical phrases (in red color) in a patent text. Important technical phrases are always near the beginning of the patent text and appear multiple times. Meanwhile, technical phrases contain much technical information that differs from traditional keywords. Our observation shows that structure information and position information of technical phrases are also important. Therefore, how to make good use of the structure and position information is a challenge.

Motivated by this, this paper proposes a new Unsupervised Technical phrase Extraction model by incorporating Structure and Position information (named UTESP) for identifying technical phrases in patents. According to the description in Boudin (2018), the relationship between phrases is very important for scoring technical phrases. The importance of a phrase is significantly related to the value of the phrase itself and the weight of linking it (Hu et al., 2018). We have improved it by adjusting the weight of the incoming edge of candidate phrases to find the technical phrase that best represents the patent text. Using position information increases the value of the candidate phrases, and using structural information increases the weight between

candidate phrases. Specifically, we analyzed the critical features of technical phrases in patents and then designed four steps to extract technical phrases: candidate generation, graph construction, candidate score, and candidate selection. In order to ensure that we have all possible technical phrases, we first need to build a huge candidate pool. Then we built three different graphs: semantic graph, structure graph, and position graph, that represent three different indicators and are used for scoring technical phrases in the next step. After the graph construction from the previous step, each technical phrase gets a different score, combined with the score of the statistical indicators, to summarize the total score. Because the number and ratio of technical phrases at different levels are different, we select different number of technical phrases according to the number of sentences to ensure the quality. Extensive experiments have been carried out on real-world patent data sets with seven competitive algorithms, which indicate that our proposed framework is significantly improved. Our contributions are as follows:

- Take advantage of structure and position information for technical phrases to improve precision. Structure information refers to adjusting the incoming edge weight of candidate phrases through the distance relations between candidate phrases. Position information represents the position relationship and frequency of technical phrases. We can find technical phrases that summarize the patent text better by giving more weight to those phrases with better positions and higher frequency.
- Propose a new unsupervised technical phrase extraction framework by incorporating structure and position information that can effectively solve the issue of technical phrase extraction from patent texts. Specifically, we first build a large enough candidate pool. Secondly, we construct three graphs that indicate three different indicators and use them to evaluate the technical phrases separately. Thirdly, we score each phrase by combining four different indicators. Finally, we choose a reasonable number of technical phrases at different levels.
- The effectiveness of our framework has been demonstrated by comparing with seven competitive algorithms on the patent datasets in terms of three evaluation metrics: Precision, Recall, and F1 scores. Besides, our new framework indicated significant improvements in the representation ability of technical phrases by comparing Information Retrieval Efficiency (IRE) with competitive algorithms.

The rest of this paper is organized as follows. In Section 2, we describe related work. Section 3 summarizes the characterization of the technical phrase. In Section 4, a new framework for technical phrase extraction is proposed. Section 5 gives the experimental analysis and result. Finally, Section 6 gives a brief conclusion.

2. Related work

2.1. Key phrase extraction

Key phrase extraction wants to extract the key information that can summarize the text, and key phrase extraction has been extensively studied by supervised and unsupervised methods (Liu et al., 2020).

For example, Xie et al. (2017) was a document-specific key phrase extraction method based on sequential pattern mining. From the experimental results, we can see that this framework can improve the precision of key phrases. Goz and Mutlu (2022) proposed a model based on node ranking. The model uses multiple graphs to model the text globally and allow multiple relationships between candidate keywords. Bougouin et al. (2013) proposed a topic model, the position relationship between topics represents the weight between topics, and then applies a graph-based ranking model to each topic. Boudin (2018) proposed a complex model. By adjusting the edge weight of candidate key phrases, the hierarchical clustering algorithm is used to cluster the more important positions at the edge of the text as the topic one by one to achieve the key phrase score. Florescu and Caragea (2017) proposed a model based on unsupervised graph, called position ranking, which combines all the position information of a word. Biswas et al. (2018) proposed a keyword extraction model, it consists of four stages: preprocessing, text graph representation, node weight assignment, and keyword extraction. Duari and Bhatnagar (2020) proposed a supervised framework for automatic keyword extraction from individual documents. They exploited the complex interactions of node attributes to design a supervised keyword extraction method.

Although key phrases are essential for text and can summarize the information of the text, they do not necessarily have a technical meaning.

2.2. Named entity recognition

Named entity recognition (NER) mainly focuses on classifying the extracted phrases into different categories. Liu et al. (2020).

For instance, Yang, Chen et al. (2020) proposed a new framework, it captures global and local features from different angles to improve NER. The model combines local and global features respectively, and then input these features into the BiLSTM-CRF (Bi-directional long-short term memory conditional random field) for sequence annotation. Mayhew et al. (2020) using a pre-training goal to predict the text and solve the robustness of NER system in noisy or uncertain case data. By appending the output distribution to the character embedding, the pre-trained real situation is combined with standard BiLSTM-CRF model. Chiu and Nichols (2016) proposed a new neural network architecture, which adopts the features of automatic recognition of word level and character level using the methods of BiLSTM (Bi-directional long-short term memory) and CNN (recurrent neural network) network. A coding method of partial word matching in neural network is proposed and compared with existing methods. Ji et al. (2022) proposed a new binding learning paradigm, it is realized by bundling the sequence tag based and cross-based NER models. The new binding learning paradigm can model NER tasks from the perspective of token and span level, and this model can also obtain the semantic information of token and span level.

Named entity recognition focuses on putting phrases of the same type together and is often used for classification, and it cannot be used for technical phrase extraction directly.

2.3. Concept extraction

Concept extraction is widely used in medical treatment, and its purpose is to find words or phrases that describe concepts (Liu et al., 2020).

Specifically, Li et al. (2018) studied concept mining, transforming unstructured information into structured information, and supporting downstream analysis tasks. A new method of mining concepts based on the context of candidate phrases is proposed. Then, they summarize the context information of candidate concepts by learning the embedding vector representation, and evaluate phrases by various complex metrics. Si et al. (2019) proposed a clinical concept extraction model, which automatically annotates clinical problems, treatments and tests

by embedding context words in specific fields. In the corpus combining clinical and relevant Wikipedia, a context word embedding model is first trained. Then, a BiLSTM-CRF model is trained with the context word embedding model for clinical concept extraction. Fang et al. (2021) proposed a new model, it uses titles, topics and clue words as a supervisor to guide concept extraction. The new model is composed of two attention networks, these two attention networks collect title and topic information to help extract each concept in a document.

There are similarities between concepts and technical phrases, but not all concepts have a technical meaning. Therefore, concept extraction cannot be directly used for technical phrase extraction.

2.4. Technical phrase extraction

The technical phrase has both the ability to summarize the patent text and a technical meaning. At present, there are few works dedicated to technical phrase extraction.

Specifically, Liu et al. (2020) focused on the issue of constructing a technical portrait for each patent and summarized patents from a technical point of view by technical phrases. By combining the features of phrases and the special structure of patent documents, namely UMTPE is established. In particular, in order to test the presentation ability of technical portraits, an emerging evaluation index Information Retrieval Efficiency (IRE) was proposed to supplement the traditional evaluation index. On the real patent data set, a large number of experiments have proved the effectiveness of the UMTPE framework.

In sum, key phrase extraction aims to extract phrases that can summarize the text. However, key phrases are not always technical phrases that contain technical information. Named entity recognition refers to classifying different types of phrases in different categories and cannot be used for technical phrase extraction. Concept extraction refers to extracting phrases with conceptual characteristics. Concepts do not necessarily have technical meanings and cannot be used for technical phrase extraction. Technical phrases not only need to find phrases with technical meanings but also have equally powerful representation abilities. Therefore, the above existing three types of methods are not suitable for technical phrase extraction.

Motivated by this, we propose a new framework to extract the technical phrases in the patent text. We first analyze the critical features of technical phrases in the patent. Then, we designed four steps to extract technical phrases named candidate generation, graph construction, candidate score, and candidate selection. In particular, we make full use of the position information and the structure information to improve the effectiveness of technical phrase extraction.

3. Characteristics of technical phrase

In order to extract technical phrases, we need to analyze the characteristics of technical phrases first. Through observation and reading of relevant literature, we found some characteristics of technical phrases. Each patent text consists of three parts: "Title" "Abstract" and "Claims", wherein "Title" mainly describes the most important words of the patent text, "Abstract" summarizes the brief description and main theme of the patent text, and "Claim" is the detailed description of the patent information. In Liu et al. (2020), the authors have summarized four characteristics as follows:

- Part of Speech: There are many parts of speech in technical phrases, noun technical phrases account for more than 90%.
- Number of Words: most of them are composed of 2–4 words, sometimes up to 5 words.
- Semantic Context: There are many similar technical phrases in the patent text, such as "neutral conductor" and "thermoelectric conductor". These technical phrases are more similar in semantics. In addition, there are also differences between technical phrases, so each technical phrase is independent.

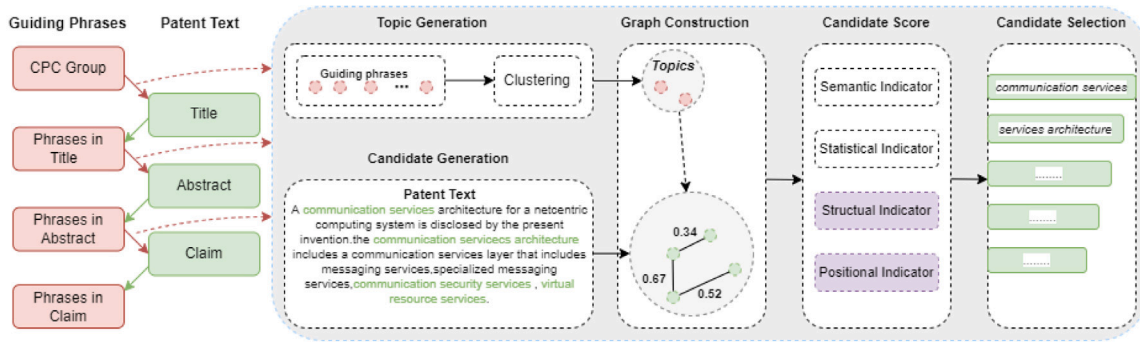


Fig. 2. Our proposed UTESP framework. Patent texts are divided into three levels from top to down: “Title”, “Abstract” and “Claim”. Meanwhile, each level is handled by the following four steps: candidate generation, graph construction, candidate score, and candidate selection. Besides, the technical phrases generated at the upper level will guide the extraction of technical phrases at the next level.

- Multi-level Architecture: The patent text can be divided into three levels: “Title” “Abstract” and “Claim”. The extracted phrase from the previous level can be used to guide the extraction of technical phrases at the next level.

However, according to expert statistics and our observations, these needed to be more comprehensive. Two critical pieces of information should have been considered: position and structure information.

3.1. Structure information

Inspired by Boudin (2018), we found that adjusting the weight of the incoming edge for the first candidate phrase of each topic helps to extract phrases that are important to the document. Therefore, we make full use of the structure relationship between phrases to help technical phrase extraction.

3.2. Position information

As shown in Fig. 1, technical phrases close to the beginning of document and appears more frequently can better represent the document, such as “radio communication devices”. Due to its specific and unique characteristics, many technical phrases appear prominently at the beginning of the patent text. The basic idea of position information is to assign greater weight (or probability) to technical phrases that appear early and frequently in patent documents (Florescu & Caragea, 2017). Specifically, position and frequency of a candidate phrase can provide more information, we can find phrases that are more important to the text. For example, if the same phrase is found in the following positions: fourth, sixth, and eighth, we can calculate its weight as: $\frac{1}{4} + \frac{1}{6} + \frac{1}{8} = \frac{13}{24}$.

Next, we will detail how to fully use structure and position information to extract technical phrases in patents.

4. Our proposed technical phrase extraction framework

Based on the current work and our new findings, we developed a new unsupervised technical phrase extraction framework (named UTESP) to identify and extract phrases with technical meaning from patent texts. As shown in Fig. 2, from top to down, patent texts are divided into three levels: “Title”, “Abstract” and “Claim”. UTESP extracts technical phrases in each level through the following four steps: Candidate Generation, Graph Construction, Candidate Score, and Candidate Selection.

In Fig. 1, we can find that technical phrases are rich in technical meaning and differ from traditional key phrases. Therefore, we can guide the extraction of technical phrases by introducing some professional corpora. The patent classification system contains a lot of technical information, which can guide the extraction of technical

phrases (Liu et al., 2020). For example, CPC Group (Cooperative Patent Classification Group), whose description (multichannel communication, wireless communication network, and so on) is highly related to technology (Liu et al., 2020). Specifically, we map the technical words and phrases in the CPC system to the embedding space and cluster it into several centroids to find the topic of several technical words and phrases. The topics formed by clustering CPC system is used to guide technical phrase recognition and extraction in “Title” of patent text, due to the correlation between different levels, the technical phrases generated at the “Title” level will guide the extraction at the “Abstract” level, the technical phrase extracted by “Abstract” is used to guide the technical phrase extraction in “Claim”.

4.1. Candidate generation

In order to improve the integrity of phrase extraction, we preprocess patent texts using various phrase extraction tools and construct a large-scale candidate phrase library. Regarding efficiency and the part of speech, we add a noun phrase extraction rule for these tools. Specifically, we apply four representative tools as follows:

- Autophrase (Shang et al., 2018): extracts significant phrases based on quality estimation and occurrence recognition.
- DBpedia (Daiber et al., 2013): can automatically annotate DBpedia resources mentioned in the text.
- Spacy (Honnibal & Montani, 2017): use the entity and noun phrase chunking part to generate candidate phrases.
- Noun Phrase Extraction (Bird et al., 2009): most technical phrases are noun phrases. To ensure all candidates are included, we use grammar tags to supplement the candidate phrase pond.

The first three tools, Autophrase, DBpedia, and Spacy, aim to extract as many phrases as possible (not all extracted phrases are technical phrases). They can help to expand the candidate phrase pool to avoid the omission of critical technical phrases. All the phrases generated by these four tools will be merged. Meanwhile, we filter out all the single words and remove the repetition. After that, we constructed a complete candidate phrase library.

4.2. Graph construction

Inspired by Liu et al. (2020), we used the skip-gram model to train unigram embedding. Then we obtain the vector of the technical phrase by averaging the dense vector of each word in the phrase (Liu et al., 2020). After that, we make each candidate phrase a node and construct graph $G = (V, E)$, where V represents nodes and E represents the edge between two nodes. We construct three graphs using different ways as follows:

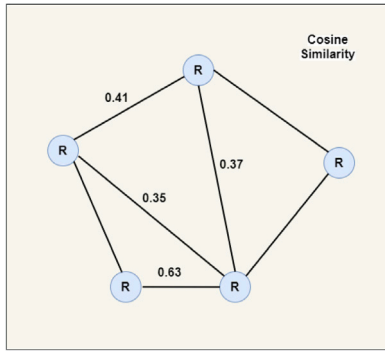


Fig. 3. Semantic graph. The node R in the graph represents the candidate phrase, and the edge weight between the nodes is the cosine similarity. Semantic graphs are constructed to score candidate phrases from a semantic perspective.

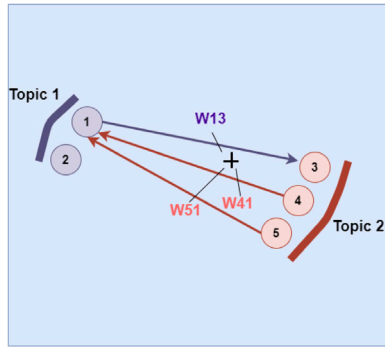


Fig. 4. Structure graph. A structure graph is a multipartite graph that uses a clustering algorithm to group similar phrases to form multiple topics. Nodes in the graph represent candidate phrases, and the weight of edges is the distance relationship between nodes. The structural graph is constructed to score candidate phrases from a structural perspective.

- A semantic graph G_R , where the weight of the edge is determined by the cosine similarity between the two nodes v_i and v_j , as shown in Fig. 3.
- A structure graph G_S , where two nodes v_i and v_j are connected based on the distance relationship between nodes in d , select the most important technical phrase of each topic by adjusting the incoming edge weight of candidate phrases through the distance relations between candidate phrases, as shown in Fig. 4.
- A position graph G_P , where two nodes v_i and v_j are connected by an edge $(v_i, v_j) \in E$ if the phrases corresponding to these nodes co-occur within a window of w in the document d , as shown in Fig. 5.

4.3. Candidate score

In this section, we design four measurement indicators from the statistical, semantic, structure, and position perspectives. Statistical information is crucial for text mining, and the design of statistical measurement indicators is intuitive (Liu et al., 2020). The semantic measurement indicators is mainly to establish the internal relationship between candidate phrases (Liu et al., 2020). The position measurement indicator focuses on the position and frequency of candidate phrases, while the structure measurement indicator reflects the importance of candidate phrases by adjusting the relationship between phrases. According to these four indicators, we can get the ranking and score of each phrase.

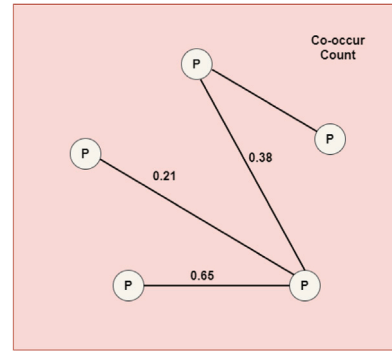


Fig. 5. Position graph. The node P in the graph represents the candidate phrase, and the edge weight between the nodes is co-occurring count within a window of w in the document d . The position graph is constructed to score candidate phrases from a position perspective.

4.3.1. Statistical measurement indicators

Statistical information is vital for phrase scoring. We use two intuitive statistical measurements: self-length and coverage, to measure statistical information (Liu et al., 2020).

Self-length calculates the length of the candidate phrase. We define the phrase length as:

$$self_length_i = \begin{cases} 1 & \text{len}(\theta_i) = 2, 3, 4 \\ 0.5 & \text{len}(\theta_i) = 5 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\text{len}(\theta_i)$ represents the number of words for the candidate phrase θ_i .

Coverage measures the number of sentences where candidate phrases are located. Technical phrases usually appear in multiple sentences, because they are essential for connecting different parts of a paragraph. Obviously, the larger the number, the more documents the phrase represents. From this perspective, we designed a formula to count the number of sentences containing candidate phrases in the patent document:

$$Coverage_i = \sum_k \mathbb{I}(\theta_i \in \text{sentence}_k) \quad (2)$$

where $\mathbb{I}()$ indicates whether the equality in brackets is true or false, θ_i represents the candidate phrase, sentence_k indicates the k th sentence. If the equality is true, the value is 1, otherwise it is 0.

4.3.2. Semantic measurement indicators

Semantic measurement indicator scores each candidate phrase from the semantic perspective (Liu et al., 2020).

Topic relations calculates the intimacy between the candidate phrase and the topics of the guiding phrase. We define the topic relationship as:

$$Topic_relation_i = \max_k \cos(\theta_i, Topic_k) \quad (3)$$

where θ_i represents the candidate phrase, $Topic_k$ is the topics formed by guiding phrases clustering.

Semantic relations measure the connectivity of technical phrases. Generally speaking, similar technical phrases often appear in the same environment. We define the semantic relation as:

$$Semantic_relation_i = \frac{\sum_{j \neq i} \mathbb{I}(\cos(\theta_i, \theta_j) \geq T)}{\sum_{j \neq i} \mathbb{I}(1)} \quad (4)$$

where $\mathbb{I}()$ indicates whether the equality in brackets is true or false, θ_i and θ_j represent the candidate phrases, and T is a threshold.

Semantic independence mainly measures the difference between technical phrases, which means that there are both correlation and

difference between technical phrases. Each technical phrase is an independent individual. We designed the following indicators to distinguish other technical phrases in the semantic graph:

$$Semantic_independence_i = \min_{j \neq i} (1 - \cos(\theta_i, \theta_j)) \quad (5)$$

where θ_i and θ_j represent the candidate phrases.

Using this node score and the edge weight, we conduct the NE-rank algorithm (Bellaachia & Al-Dhelaan, 2012) on the semantic graph G_R . Ne-rank is significantly improved compared with PageRank (Brin & Page, 1998) and TextRank (Mihalcea & Tarau, 2004). PageRank is an algorithm used by Google Search to rank web pages in their search engine results. PageRank counts the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites will likely receive more links from other websites. TextRank algorithm is a graph-based ranking model for text processing, where the vertices represent phrases and edges represent relationships between phrases. When one vertex links to another one, it votes for that other vertex. The higher the number of votes cast for a vertex, the higher the importance of the vertex. NE-rank not only considers the weight of candidate phrases, but also considers the relationship between phrases on the semantic graph. By summing the scores from Eqs. (1) to (5), we can get the scores of each node, and then normalized to get S_i . The statistical and semantic score of a node can be obtained algebraically by recursively computing the following equation:

$$Score_{sem}(v_i) = (1 - \beta) \cdot S_i + \beta \cdot S_i \cdot \sum_{v_j \in Adj(v_i)} \frac{w_{ji}}{O(v_j)} Score_{sem}(v_j) \quad (6)$$

where $O(v_j) = \sum_{v_k \in Adj(v_j)} w_{jk}$, and S_i is the normalized score of node v_i . Judging from the best experimental results, β is set to 0.85.

4.3.3. Structure measurement indicator

In this section, besides the statistical and semantic information, we designed an improved measurement indicator to score each candidate phrase from the structure perspective.

The structural score measures the importance of a phrase by adjusting incoming edge weights. We build a multipartite graph for a document in terms of the approaches proposed by Boudin (2018). Then, the hierarchical agglomerative clustering (HAC) algorithm puts some similar candidate phrases into the same topic and build the multipart graph G_S , the graph is composed of multiple topics, and only candidate phrases between different topics are linked into edges. The weights of edges between node v_i and node v_j can be defined as follows:

$$w_{ij} = \sum_{p_i \in P(\theta_i)} \sum_{p_j \in P(\theta_j)} \frac{1}{|p_i - p_j|} \quad (7)$$

where θ_i and θ_j are corresponding candidate phrases, p_i is the position of the phrase θ_i , and $P(\theta_i)$ is all the positions of the phrase θ_i .

The order of candidate can provide more information and can be used for weight adjustment. Adjusting the incoming edge weights for the first occurring candidate of each topic is helpful for the important phrase extraction (Boudin, 2018). Inspired by this, our new method adjusts the incoming edge weight according to the position of the candidate phrase. The weight adjustment formula of edges is as follows:

$$w_{ij} = w_{ij} + \varepsilon \cdot T_i \cdot \exp^{\frac{1}{p_i}} \cdot \sum_{c_k \in T(\theta_j) \setminus \{\theta_j\}} w_{ki} \quad (8)$$

where w_{ij} is the weight of the edges between two different nodes, $T(\theta_j)$ refers to all candidate phrases in the same topic as θ_j , and ε is a hyperparameter that controls the degree of adjustment, T_i refers to the value of topic relevance in Eq. (3).

We calculate the candidate phrase score by using the TextRank algorithm (Mihalcea & Tarau, 2004). TextRank algorithm uses the topic information recursively calculated in the structure graph. The structure score of candidate c_i is calculated as follows:

$$Score_{str}(\theta_i) = (1 - \lambda) + \lambda \cdot T_i \cdot \sum_{c_j \in I(\theta_i)} \frac{w_{ij} Score_{str}(\theta_j)}{\sum_{c_k \in O(\theta_i)} w_{jk}} \quad (9)$$

where $I(\theta_i)$ consists of all predecessors of θ_i , $O(\theta_j)$ consists of all successors of θ_j , and in order to make the nodes in the graph jump to other nodes with probability, λ was designed as damping factor, T_i refers to the value of topic relevance in Eq. (3). According to relevant experiments, λ is set to 0.85.

After iterating through the graph sorting algorithm until convergence, we can get the ranking list and each candidate phrase score $Score_{str}$.

4.3.4. Position measurement indicator

Based on the observations in Section 3, we design a new intuitive position information measure.

Generally, the weight of the edges between node v_i and node v_j is based on the co-occurrence count of two phrases in the window w of continuous labels in the document d . Formally, let G_p be an undirected graph constructed as above and let M be its adjacency matrix. If there is a relationship between two nodes, set it as the weight of edge $m_{ij} \in M$, otherwise set it to 0.

Let S represent the vector of $v_i \in V$ PageRank scores. In step $t + 1$, the PageRank score of each node can be calculated recursively as Florescu and Caragea (2017):

$$S(t+1) = \widetilde{M} \cdot S(t) \quad (10)$$

where \widetilde{M} refers to normalized form of M .

Because PageRank will fall into the circular process of the graph, we added a parameter α so that we can operate on other nodes in the graph. Therefore, the calculation of S becomes (Florescu & Caragea, 2017):

$$S = \alpha \cdot \widetilde{M} \cdot S + (1 - \alpha) \cdot \widetilde{p} \quad (11)$$

where \widetilde{p} refers to normalized form of p . The vector \widetilde{p} is initialized, it is equal, which means that the probability of jumping from one node to any node is equal. When \widetilde{p} is not the same, random walk is more likely to be at nodes with higher probability in the graph.

The position score of a node $S(v_i)$ can be obtained algebraically by recursively computing the following equation:

$$Score_{pos}(v_i) = (1 - \alpha) \cdot \widetilde{p}_i \cdot T_i + \alpha \cdot \sum_{v_j \in Adj(v_i)} \frac{w_{ji}}{O(v_j)} Score_{pos}(v_j) \quad (12)$$

where $O(v_j) = \sum_{v_k \in Adj(v_j)} w_{jk}$ and \widetilde{p}_i is the normalized position weight of candidate phrase for node v_i , T_i refers to the value of topic relevance in Eq. (3). Based on experiments and analysis, we found that $\alpha = 0.85$ is the best choice.

The PageRank scores are recursively computed until convergence, we can get the score $Score_{pos}$ and ranking list of each candidate phrase.

Through the above three indicators, we can get three different scores for each candidate phrase, and then the average score can finally be used for the score and ranking of each phrase as:

$$Score(v_i) = \frac{Score_{sem}(v_i) + Score_{str}(v_i) + Score_{pos}(v_i)}{3} \quad (13)$$

4.4. Candidate selection

After scoring in the previous step, we select the top K candidate phrases with the highest scores. Because the length of patent text at different levels in different documents is different, the number of technical phrases is also different, we set different ratios for $K/N_{sentence}$ in the title, abstract, and claim, we define $K/N_{sentence}$ according to the number of sentences ($N_{sentence}$) as follows:

$$\frac{K}{N_{sentence}} \approx \begin{cases} 1 \sim 2 & \text{Title} \\ 2 & \text{Abstract} \\ 1 & \text{Claim} \end{cases} \quad (14)$$

Algorithm 1: Unsupervised Technical phrase Extraction by incorporating Structure and Position information (UTESP)

- 1 **Input:**
 - 2 Q : the CPC group phrase set;
 - 3 **Output:**
 - 4 S : the selected technical phrases set;
 - 1: **Initialization:** $S = \{\}$;
 - 2: **Repeat from Title, Abstract to Claim**
 - 3: Candidate generation: extract phrases by Autophrase, DBpedia, Spacy, and Noun Phrase Extraction and remove the repetition;
 - 4: Graph construction: build the semantic graph G_R , the structure graph G_S , and the position graph G_P by embedding;
 - 5: Candidate score: get the scores ($Score_{sem}$, $Score_{str}$, $Score_{pos}$) of each phrase through the graph ranking algorithm on graph G_R , G_S and G_P respectively;
 - 6: Candidate selection: rank and select a certain number of technical phrases by the average scores into S ;
 - 7: New topics are formed by clustering the technical phrases selected in S to guide the subsequent extraction of technical phrases;
 - 8: **After** three cycles, remove the repeated technical phrases;
 - 9: **Output** S .
-

The details our new proposed framework is shown as Algorithm 1. Specifically, candidate generation in step 3 aims to build a large candidate pool for us. Step 4 constructs three different graphs to fully reflect the characteristics of technical phrases to meet different needs. Step 5 uses different graph ranking algorithms to generate the ranking and score of each technical phrase. Candidate selection in step 6 reasonably selects different ratios of technical phrases generated for different patent parts. With the selected technical phrases in the title, new topics are formed by clustering the technical phrases selected in the current cycle to guide the subsequent extraction of technical phrases in the abstract. Then the selected technical phrases in the abstract will guide the extraction of technical phrases in the claim. Finally, output non-repetitive technical phrases in S .

5. Experiments

5.1. Experimental setup

5.1.1. Datasets

This paper has carried out experiments on patent datasets in the fields of electrical and mechanical engineering (Liu et al., 2020). The former involves patents relating to mechanical engineering, lighting, heating, weapons, and blasting engines or pumps, while the latter is related to the electric field. We randomly sample 1000 pieces of patent data in the Mechanical Engineering and Electricity datasets, respectively. The details of the datasets are shown in the Table 1.

5.1.2. Implementation details

Our UTESP framework is implemented in PYTHON. Besides, there are some hyperparameters for UTESP as follows:

- The hyperparameters of guiding phrase clustering: According to the experiment in Liu et al. (2020), in the guidance phrase, the minimum cluster size in CPC system is 3, and the cluster size in “Title” and “Abstract” is 100.
- The hyperparameters of candidate score: According to the results of many experiments, we set $T = 0.5$ in Eq. (4) and $\beta = 0.85$ in Eq. (6) as the empirical values. In terms of the experiments of Boudin (2018), we set $\varepsilon = 1.1$ in Eq. (8) for the structural measurement indicator, and $\lambda = 0.85$ in Eq. (9). For the position measurement indicator, we set $\alpha = 0.85$ for Eqs. (11) and (12) as the empirical values.

All experiments were run on one Tesla K80 GPU and 16 Intel CPUs.

Table 1

The statistics of two real-world patent datasets.

Dataset	Num. of patents	Avg. sentences of title	Avg. sentences of abstract	Avg. sentences of claim
Electricity	1000	1.0	3.85	13.58
Mechanical engineering	1000	1.0	3.89	16.58

5.1.3. Baselines

We compare UTESP with seven excellent methods, as described below:

- UMTPE (Liu et al., 2020) is an unsupervised technical phrase extraction framework by combining the features of technical phrases and the special structure of patent documents.
- Rake (Rose et al., 2010) extracts key phrases based on the importance of phrases and the relationship between phrases. Rake uses a set of intuitive parameters and uses these parameters to automatically extract key phrases, making them applicable to many documents and collections.
- Spacy (Honnibal & Montani, 2017) first provides pre-training, and so far supports tokenization and training in more than 60 languages. Spacy has extremely fast speed and excellent neural network model, which is used for marking, parsing, multi-task learning using pre-trained converters such as BERT. In addition to the above, it also has a complete training system and excellent model packaging.
- NE-rank (Bellaachia & Al-Dhelaan, 2012) is used for extracting phrases that take into account word weight when calculating the ranking. The purpose of NE-rank is to extract the topic key phrase that will represent the text in the tweet.
- Autophrase (Shang et al., 2018) is a phrase extraction framework, which provides two emerging technologies: robust forward distance training only and POS guided phrase segmentation combined with part of speech tags.
- DBpedia (Daiber et al., 2013) is an exceptional example of the Semantic Web application, it is also one of the largest multidomain ontologies in the world and part of Linked Data.
- ECON (Li et al., 2018) is a new method to mine concepts by learning the embedding vector representation, which summarizes the context information of each possible candidate object, and uses these embedding to evaluate the global quality of concepts.

Autophrase, DBpedia, and Spacy are three classic algorithms that have experienced the baptism of time and are recognized by many researchers. These three algorithms can extract phrases from the text, but these phrases are not all technical phrases. We compare our new algorithm with these three classic algorithms to validate the particularity of technical phrase extraction. NE-rank and Rake are effective key phrase extraction models. Through comparison, it can be found that the model used for key phrase extraction cannot be directly used for extracting patent technical phrases, which reflects the differences between key phrases and technical phrases. ECON algorithm is used for concept extraction, which does not necessarily have technical meaning, and the comparison shows the differences between concept phrases and technical phrases. UTMPE algorithm is specially designed for technical phrase extraction. However, UTMPE needs to be more comprehensive and consider the structural information and position information of technical phrases. Comparing with UTMPE can reflect the improvement of our new algorithm.

5.2. Results

5.2.1. Overall performance evaluation

In this part, we use three widely used indicators (Precision, Recall, and F1-score) to evaluate the overall performance of these eight competitive algorithms on 100 labeled patents for each dataset (Hasan &

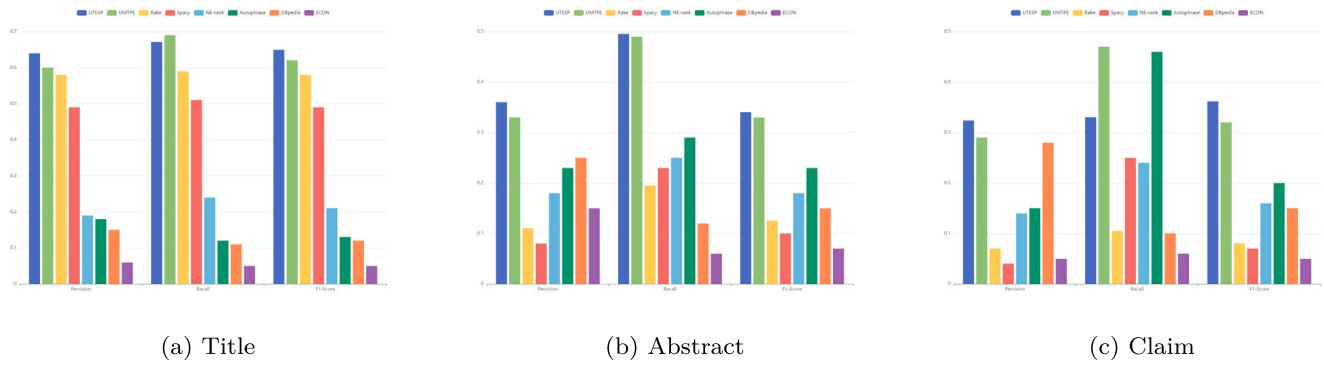


Fig. 6. Evaluation on Dataset Electricity, the performance of Precision(P), Recall(R), and F1-score(F1) for these eight competing algorithms on Title, Abstract, and Claim, respectively.

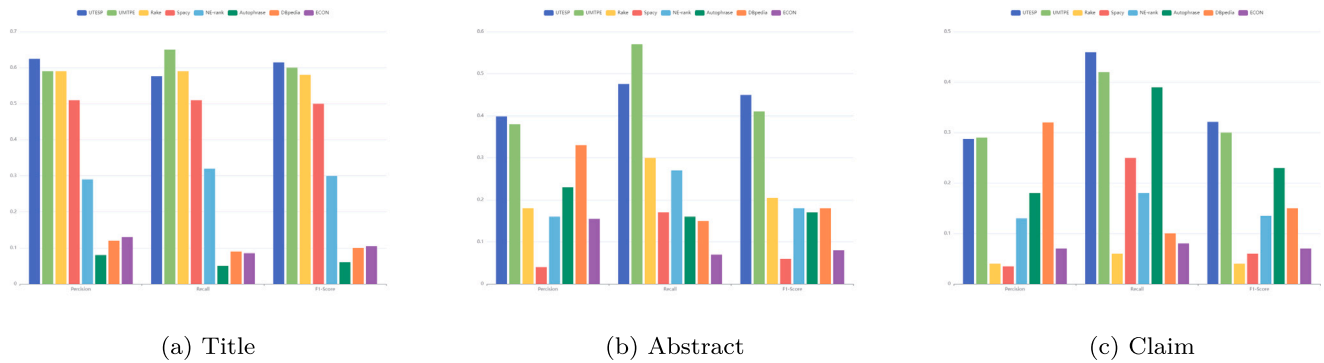


Fig. 7. Evaluation on Dataset Mechanical Engineering, the performance of Precision(P), Recall(R), and F1-score(F1) for these eight competing algorithms on Title, Abstract, and Claim, respectively.

Table 2

Performance of Precision(P), Recall(R), and F1-score(F1) for these eight competing algorithms on Two patent datasets.

Method	Electricity			Mechanical engineering		
	P	R	F1	P	R	F1
UTESP	0.4412	0.4857	0.4503	0.4537	0.5085	0.4619
UMTPE	0.4104	0.5521	0.4285	0.4153	0.5389	0.4337
Rake	0.2525	0.3000	0.2564	0.2693	0.3154	0.2733
Spacy	0.2017	0.3289	0.2201	0.1999	0.3186	0.2116
NE-Rank	0.1680	0.2422	0.1778	0.1908	0.2620	0.2034
Autophrase	0.1825	0.2911	0.1906	0.1653	0.1937	0.1487
DBpedia	0.2266	0.1091	0.1356	0.2616	0.1096	0.1399
ECON	0.0892	0.0574	0.0607	0.1236	0.0834	0.0906

Ng, 2014; Liu, Ge et al., 2011; Wang et al., 2018). We get the original state of each word in the extracted phrases and label phrases, then calculate the result (Liu et al., 2020). Table 2 shows the results of the three indicators (P, R, F1) for these competing algorithms. Meanwhile, Figs. 6 and 7 indicate the performance of these competing algorithms on the title, abstract and claim of patents.

To verify whether the performance of UTESP and its competitors is significantly different, we performed the Friedman test at 95% significance level under the null hypothesis (Demšar, 2006). If the null hypothesis is rejected, there is a significant difference in the performance of OHSFS and its competitors. When the null hypothesis of the Friedman test was rejected, we proceeded to the Nemenyi test as a post-hoc test (Demšar, 2006). The p-values of precision, recall, and F1-score are 1.3865e-07, 1.3865e-07, and 1.3865e-07, respectively. Thus, there is a significant difference in performance among these competing algorithms. According to the Nemenyi test, the value of CD is 7.4305. Fig. 8 shows the statistical test of these competing algorithms in Precision, Recall, and F1-score cases.

From Table 2 and Figs. 6 to 8, we can observe that:

- UTESP vs. UMTPE: Although UMTPE described the technical phrase and pointed out some characteristics, it did not consider the importance of the position of the technical phrase. Meanwhile, UMTPE did not fully use the relationship between the technical phrases. Therefore, UTESP performs better than UMTPE in cases of Precision and F1. For UTESP considers more candidate phrases, UMTPE gets a higher value of recall than UTESP on these two datasets.
- UTESP vs. Rake: UTESP performs better than Rake in cases of precision, recall, and F1. Rake is used for keyphrase extraction and only considers the frequency and degree of words. However, key phrases are only sometimes technical phrases. Although Rake is fast, it is not suitable for technical phrase extraction.
- UTESP vs. Spacy: Spacy has many functions and supports multiple languages. However, Spacy cannot be used for technical phrase extraction because it does not distinguish between technical phrases and common phrases. On the contrary, UTESP is specially designed for technical phrase extraction to perform better.
- UTESP vs. Ne-rank: UTESP performs much better than NE-rank in precision, recall, and F1 cases. Like Spacy, NE-rank aims to extract topical key phrases representing topics in tweets and does not distinguish between technical phrases and key phrases.
- UTESP vs. Autophrase: Autophrase is a semi-supervised phrase extraction method and can adapt to various fields with only a small amount of human resources. Autophrase uses high-quality phrases in the existing general knowledge base (such as wiki) and distance training (reduce human input), and POS-guided phrase segmentation (improve model performance). However, Autophrase also cannot distinguish between technical phrases and common phrases, so it cannot be directly used for technical phrase extraction.

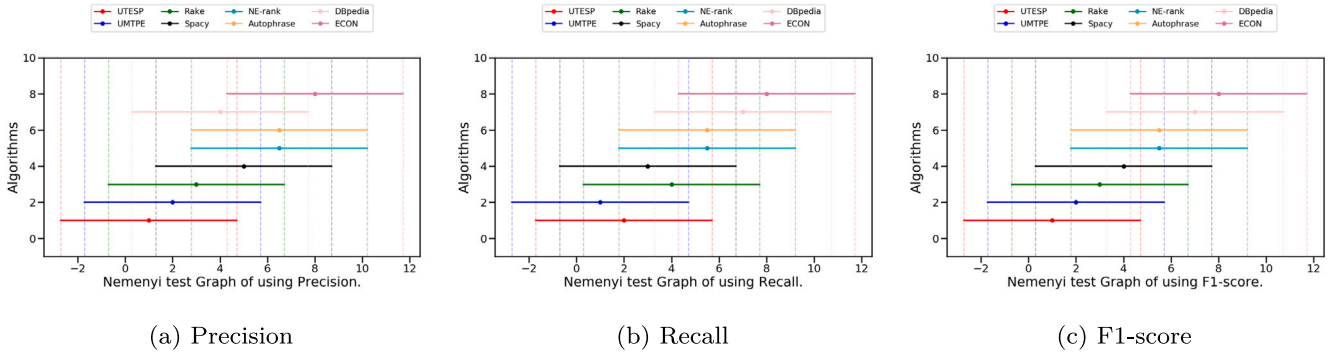


Fig. 8. The statistical test graph of these eight competing algorithms on Precision, Recall, and F1-score.

- UTESP vs. DBpedia: DBpedia is a widespread phrase extraction tool, and its primary role is to enhance Wikipedia’s search function and link other data sets to Wikipedia. Therefore, DBpedia cannot be directly used for technical phrase extraction, and UTESP performs much better than it.
- UTESP vs. ECON: ECON performs worst among all these competing algorithms. ECON is designed for concept mining based on their occurrence contexts by learning embedding vector representations. Therefore, ECON is not suitable for technical phrase extraction.

From Table 2 and Figs. 6 to 8, we can see that the Precision and F1-score have been relatively improved, but the Recall score is relatively low. By analyzing the characteristics of technical phrases in the patent datasets. We find that the relationship between phrases with different technical meanings is too distant. From Eq. (8), we can see that the structure score is not only related to w_{ij} , is also related to T_i . When the value of w_{ij} is small, the structure score is mainly related to T_i . When the datasets is rich in technical phrases with multiple meanings, the selected phrases are closely related to T_i and are not closely related to T_i technical phrases will be missed, resulting in lower Recall scores.

In sum, Rake, Spacy, Ne-rank, Autophrase, DBpedia, and ECON are not explicitly designed for technical phrase extraction, which leads to poor performance. UMTPE is proposed to extract technical phrases, but it does not consider position and structure information of technical phrases. Therefore, our new proposed method, UTESP, is the best among these competing algorithms.

5.2.2. Representation evaluation

The extracted technical phrases can form a technical portrait of the patent text, describing the important and unique technical information of the patent text. The extraction results can be effectively verified through Information Retrieval Efficiency (IRE) (Liu et al., 2020). We evaluated and analyzed 1000 patent documents, including 100 labeled patent texts and 900 unlabeled patent texts. For these extracted technical phrases, we use them as queries to rank all patent texts.

In order to escape the influence of fewer extracted phrases, by designing PF , and modifying the score:

$$PF = \begin{cases} 1 & r \leq p \\ e^{1-r/p} & r > p \end{cases} \quad (15)$$

where r represents the number of technical phrases manually marked in the patent text, and p represents the number of technical phrases extracted by the UTESP framework.

$$\text{score}_{\text{revise}} = PF \cdot \text{score}. \quad (16)$$

Finally, if the patent text in which the technical phrase is located appears in the top 10, the score for the technical phrase is 1, otherwise it is 0. By summing and averaging the scores of each technical phrase, the extracted technical phrase representation ability score is obtained,

Table 3

Performance of representation capability (IRE) of these eight competing algorithms on Abstract and Claim.

Method	Electricity		Mechanical engineering	
	Abstract	Claim	Abstract	Claim
UTESP	0.5778	0.5213	0.5562	0.4854
UMTPE	0.5516	0.5038	0.5524	0.4457
Rake	0.6478	0.4889	0.6464	0.4115
Spacy	0.4573	0.2956	0.4063	0.2716
NE-Rank	0.4410	0.3478	0.4129	0.2797
Autophrase	0.4322	0.2733	0.2836	0.2797
DBpedia	0.2083	0.1145	0.1947	0.0865
ECON	0.1529	0.1549	0.2047	0.1710

then get the final score. The results of the representation evaluation of these seven competing algorithms on the level of “Abstract” and “Claim” are shown in Table 3.

From Table 3, we can find that UTESP gets the highest value of IRE on the “Claim” level on these two datasets but is lower than Rake on the “Abstract” level. According to the discovery and description in Liu et al. (2020), Rake can extract more long technical phrases than UTESP, longer phrases can represent more information, and the text length of “Abstract” is very short. Thus, longer phrases extracted by Rake perform better than UTESP in the “Abstract” level. However, in the “Claim” level, UTESP has improved to a certain extent. In sum, the phrases extracted by UTESP have significant advantages in representing technical information.

5.3. Case study

In this section, we conducted a case study to explain our results of UTESP framework further.

In Fig. 9, “Matching Phrase” refers to the same phrase extracted by the UTESP framework and human tag, while “Reference Phrase” refers to the phrase extracted only by the human tag. “Predicted Phrase” is the phrase that is extracted only by UTESP. From this figure, we can find that UTESP can accurately identify the technical phrases in “Title” and “Abstract”, for example “communication service architecture”. Meanwhile, we can see that most matching phrases appear near the beginning of the patent text, thus verifying the importance of position information.

6. Conclusions

With the explosively increased number of patent applications, patent mining has become increasingly important, which can bring considerable benefits to enterprises and the country. Technical phrase extraction can establish a technical portrait for each patent and summarize the technical information of the patent text from a technical

Matching Phrase	Predicted Phrase	Reference Phrase	Non-technical Phrase
Title : communication service architectures for netcentric computing systems			
Abstract : a communication services architecture for a netcentric computing system is disclosed by the present invention . the communication services architecture includes a communication services layer that includes communication security services , virtual resource services and directory services . in addition , the communication services architecture includes a transport services layer that includes message transport services , packet forwarding/internetworking services , circuit switching services , transport security services , network address allocation services and quality of service services . a network media services layer is provided that includes media access services and physical media services .			

Fig. 9. A case study of the extracted technical phrases. Matching phrase (red) refers to the same phrase extracted by both UTESP and human tag. Predicted phrase (green) and Reference phrase (purple) refer to phrases extracted only by human tag and UTESP, respectively. Blue indicates non-technical phrases.

perspective. However, technical phrase extraction is more challenging than key phrase extraction. Technical phrases must contain rich technical information and are essential to the entire patent text from a technical perspective. This paper presented a novel unsupervised framework (UTESP) for technical phrase extraction from a patent that leverages position and structure information. Patent texts are divided into “Title”, “Abstract” and “Claim”, while UTESP extracts technical phrases from each level. UTESP consists of four main steps: candidate generation, graph construction, candidate score, and candidate selection. UTESP achieves the position information of technical phrases by incorporating the position and frequency in a document into a graph ranking algorithm and achieves the structure information of technical phrases by adjusting the relationship between candidate phrases. We conduct extensive experiments with seven competing algorithms on two real-world patent datasets. The metrics we used to evaluate the performance of these eight competing algorithms include Precision (P), Recall (R), F1-score(F1) and Information Retrieval Efficiency (IRE). Our experimental results indicate that our proposed framework can achieve better results than strong baselines, with relatively significant performance improvements.

In general, patent texts are more complex than ordinary texts. From a technical perspective, the technical significance of ordinary text cannot be compared with patent text. Although this work has detailed descriptions of technical phrases and designed indicators for scoring technical phrases, we only focus on the technical phrases themselves and the relationships between technical phrases. In other words, we do not consider the guiding significance of the technical information in patent texts for technical phrase extraction. Therefore, in future work, it is crucial to establish the relationship between technical phrases and patent text.

CRedit authorship contribution statement

Peng Zhou: Conceptualization, Methodology, Writing – review & editing, Funding acquisition. **Xin Jiang**: Software, Validation, Investigation, Writing – original draft. **Shu Zhao**: Formal analysis, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under grants (62376001, 61906056, 61806002), the Natural Science Foundation of Anhui Province of China under grant 2308085MF215.

References

- Akbik, A., Bergmann, T., & Vollgraf, R. (2019). Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 724–728).
- Bellaachia, A., & Al-Dhelaan, M. (2012). Ne-rank: A novel graph-based keyphrase extraction in twitter. In *2012 IEEE/WIC/ACM international conferences on web intelligence and intelligent agent technology, volume 1* (pp. 372–379). IEEE.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc..
- Biswas, S. K., Bordoloi, M., & Shreya, J. (2018). A graph based keyword extraction model using collective node weight. *Expert Systems with Applications*, 97, 51–59.
- Boudin, F. (2018). Unsupervised keyphrase extraction with multipartite graphs. arXiv preprint arXiv:1803.08721.
- Bougouin, A., Boudin, F., & Daille, B. (2013). Topicrank: Graph-based topic ranking for keyphrase extraction. In *International joint conference on natural language processing (IJCNLP)* (pp. 543–551).
- Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7), 107–117.
- Chiu, J. P., & Nichols, E. (2016). Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4, 357–370.
- Daiber, J., Jakob, M., Hokamp, C., & Mendes, P. N. (2013). Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th international conference on semantic systems* (pp. 121–124).
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7, 1–30.
- Duari, S., & Bhatnagar, V. (2020). Complex network based supervised keyword extractor. *Expert Systems with Applications*, 140, Article 112876.
- Fang, S., Huang, Z., He, M., Tong, S., Huang, X., Liu, Y., Huang, J., & Liu, Q. (2021). Guided attention network for concept extraction. In *IJCAI* (pp. 1449–1455).
- Florescu, C., & Caragea, C. (2017). Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 1105–1115).
- Goz, F., & Mutlu, A. (2022). Mgrank: A keyword extraction system based on multigraph GoW model and novel edge weighting procedure. *Knowledge-Based Systems*, 251, Article 109292.
- Hasan, K. S., & Ng, V. (2014). Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 1262–1273).
- Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1), 411–420.
- Hu, J., Li, S., Yao, Y., Yu, L., Yang, G., & Hu, J. (2018). Patent keyword extraction algorithm based on distributed representation for patent classification. *Entropy*, 20(2), 104.
- Ji, B., Xie, Y., Yu, J., Li, S., Ma, J., Ji, Y., & Liu, H. (2022). A novel bundling learning paradigm for named entity recognition. *Knowledge-Based Systems*, 248, Article 108825.
- Li, T., Hu, L., Li, H., Sun, C., Li, S., & Chi, L. (2021). TripleRank: An unsupervised keyphrase extraction algorithm. *Knowledge-Based Systems*, 219, Article 106846.
- Li, K., Zha, H., Su, Y., & Yan, X. (2018). Concept mining via embedding. In *2018 IEEE international conference on data mining (ICDM)* (pp. 267–276). IEEE.
- Liu, Q., Ge, Y., Li, Z., Chen, E., & Xiong, H. (2011). Personalized travel package recommendation. In *2011 IEEE 11th international conference on data mining* (pp. 407–416). IEEE.
- Liu, Y., Hseuh, P.-y., Lawrence, R., Meliksetian, S., Perlich, C., & Veen, A. (2011). Latent graphical models for quantifying and predicting patent quality. In *Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1145–1153).
- Liu, Y., Wu, H., Huang, Z., Wang, H., Ma, J., Liu, Q., Chen, E., Tao, H., & Rui, K. (2020). Technical phrase extraction for patent mining: A multi-level approach. In *2020 IEEE international conference on data mining (ICDM)* (pp. 1142–1147). IEEE.

- Liu, Q., Wu, H., Ye, Y., Zhao, H., Liu, C., & Du, D. (2018). Patent litigation prediction: A convolutional tensor factorization approach. In *IJCAI* (pp. 5052–5059).
- Mayhew, S., Nitish, G., & Roth, D. (2020). Robust named entity recognition with truecasing pretraining. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34* (pp. 8480–8487).
- Mihalcea, R., & Tarau, P. (2004). Texttrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 404–411).
- Papagiannopoulou, E., & Tsoumakas, G. (2020). A review of keyphrase extraction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(2), Article e1339.
- Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. In *Text mining: applications and theory* (pp. 1–20). Wiley Online Library.
- Shang, J., Liu, J., Jiang, M., Ren, X., Voss, C. R., & Han, J. (2018). Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10), 1825–1837.
- Si, Y., Wang, J., Xu, H., & Roberts, K. (2019). Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, 26(11), 1297–1304.
- Wang, H., Chen, E., Liu, Q., Xu, T., Du, D., Su, W., & Zhang, X. (2018). A united approach to learning sparse attributed network embedding. In *2018 IEEE international conference on data mining (ICDM)* (pp. 557–566). IEEE.
- Wu, H., Zhang, K., Lv, G., Liu, Q., Yu, R., Zhao, W., Chen, E., & Ma, J. (2019). Deep technology tracing for high-tech companies. In *2019 IEEE international conference on data mining (ICDM)* (pp. 1396–1401). IEEE.
- Xie, F., Wu, X., & Zhu, X. (2017). Efficient sequential pattern mining with wildcards for keyphrase extraction. *Knowledge-Based Systems*, 115, 27–39.
- Yang, X., Bian, J., Hogan, W. R., & Wu, Y. (2020). Clinical concept extraction using transformers. *Journal of the American Medical Informatics Association*, 27(12), 1935–1942.
- Yang, Z., Chen, H., Zhang, J., Ma, J., & Chang, Y. (2020). Attention-based multi-level feature fusion for named entity recognition. In *International joint conference on artificial intelligence*.
- Yu, J., Bohnet, B., & Poesio, M. (2020). Named entity recognition as dependency parsing. arXiv preprint arXiv:2005.07150.
- Yu, Y., & Ng, V. (2018). Wikirank: Improving keyphrase extraction based on background knowledge. arXiv preprint arXiv:1803.09000.
- Zhang, L., Li, L., & Li, T. (2015). Patent mining: a survey. *ACM SIGKDD Explorations Newsletter*, 16(2), 1–19.
- Zhang, L., Li, L., Li, T., & Zhang, Q. (2014). Patentline: analyzing technology evolution on multi-view patent graphs. In *Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval* (pp. 1095–1098).