Technical Phrase Extraction for Patent Mining: A Multi-level Approach

Ye Liu, Han Wu, Zhenya Huang, Hao Wang, Jianhui Ma, Qi Liu, Enhong Chen*, Hanqing Tao, Ke Rui
Anhui Province Key Laboratory of Big Data Analysis and Application, School of Data Science &
School of Computer Science and Technology, University of Science and Technology of China
{liuyer, wuhanhan, wanghao3, hqtao, kerui}@mail.ustc.edu.cn, {huangzhy, jianhui, qiliuql, cheneh}@ustc.edu.cn

***Abstract*—Recent years have witnessed a booming increase of patent applications, which provides an open chance for revealing the inner law of innovation, but in the meantime, puts forward higher requirements on patent mining techniques. Considering that patent mining highly relies on patent document analysis, this paper makes a focused study on constructing a technology portrait for each patent, i.e., to recognize technical phrases concerned in it, which can summarize and represent patents from a technology angle. To this end, we first give a clear and detailed description about technical phrases in patents based on various prior works and analyses. Then, combining characteristics of technical phrases and multi-level structures of patent documents, we develop an Unsupervised Multi-level Technical Phrase Extraction (UMTPE) model. Particularly, a novel evaluation metric called Information Retrieval Efficiency (IRE) is designed to evaluate the extracted phrases from a new perspective, which greatly supplements traditional metrics like Precision and Recall. Finally, extensive experiments on real-world patent data show the effectiveness of our UMTPE model.**

***Index Terms*—Technology Portrait, Technical Phrase Extraction, Patent Mining, Multi-level**

I. Introduction

According to statistics of WIPO (World Intellectual Property Organization)[1], patent applications keep growing worldwide every year since 2004 (except 2009). This explosive growth indeed brings a valuable data basis for revealing the inner law of innovation [1]–[4], but at the same time, puts forward higher requirements on patent mining techniques [5].

As a matter of fact, patent mining often highly relies on text analysis, i.e., how to process, quantify and analyze key information of patent documents [6], [7]. An effective step here is to construct a technology portrait for each patent, that is, to identify technical phrases involved, and thus summarize its key information from a technical perspective. For example, one given patent may contain "wireless communication" and "multiplex communication", whose repeated occurrence indicates this patent is closely related to "electric network" and might be a new innovation about "multiplex wireless communication". If we extract all phrases like that, the combination of them can be seen as a technology portrait tagging this patent. Several examples of *technical phrases* are listed in TABLE I and a more clear description can be found in Section III.

TABLE I: Technical Phrases vs. Non-technical Phrases

Domain	Technical phrase	Non-technical phrase
Electricity	wireless communication, netcentric computer service	wire and cable, TV signal, power plug
Mechanical Engineering	fluid leak detection, power transmission	building materials, steer column, seat back

As far as we are concerned, there have been few works specially designed for technical phrase extraction, while some relevant works have been explored on phrase extraction. According to the extraction target, they can be divided into *Key Phrase Extraction* [8], *Named Entity Recognition (NER)* [9] and *Concept Extraction* [10]. Key Phrase Extraction aims to extract phrases that provide a concise summary of a document, which prefers those both frequently-occurring and closed to main topics. NER focuses on locating and classifying named entities into pre-defined categories. Concept Extraction is the closest to technical phrase extraction, which aims to find words or phrases describing a concept. However, they are obviously different from technical phrase extraction as phrases like "user preference" belong to concepts but not technical phrases.

Indeed, there are many technical and domain challenges inherent in designing effective solutions to this problem. First, as the technology meaning of texts is hard to quantify, there are more perplexing and unreachable characteristics among technical phrases. Second, one patent document often contains "Title", "Abstract" and "Claim", serving as an organic combination of multi-levels and show strong connections. How to combine information from different levels and effectively utilize their relations are also key challenges. Third, traditional evaluation metrics like Precision and Recall are relatively one-sided, and we need an evaluation method specially designed for technical phrases to improve the confidence of evaluation.

To address above challenges, in this paper, we propose an Unsupervised Multi-level Technical Phrase Extraction (UMTPE) model for recognizing technical phrases in patents. Specifically, we analyze key characteristics of technical phrases in patents and design several measurement indicators for them from both semantic and statistical angles. Then, considering the relations between different levels in patents, we further design components (i.e., Topic Generation, Topic Relevance) to relate adjacent levels, which could utilize the implied information in multi-level structures extensively. Finally, Information Retrieval Efficiency (IRE) is designed to supplement traditional evaluation metrics, which could evaluate extracted technical phrases from the perspective of representation ability. Extensive experiments on real-world patent data show the effectiveness of UMTPE model.

* denotes the corresponding author

[1] https://www.wipo.int/ipstats/en/

2374-8486/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDM50108.2020.00139

Our code of UMTPE is available at https://github.com/bigdata-ustc/UMTPE.

II. Related Work

A. Key Phrase Extraction

Key Phrase Extraction aims to extract phrases that provide a concise summary of a document, which has been widely studied with supervised [11], [12] and unsupervised methods [13], [14]. For one thing, supervised methods often target at training a complicated model with the help of labeled data or external knowledge base. For instance, Meng et al. [11] designed an encoder-decoder framework to generate keyphrases from original text. For another, unsupervised methods focus on mining the inner-connections in documents in response to lack of labeled data. In this term, Bellaachia et al. [14] designed a ranking algorithm to evaluate the importance of words in documents, from which they further formulated key phrases.

B. Named Entity Recognition

NER focuses on locating and classifying named entities into pre-defined categories, which is often regarded as a sequence problem and tackled by RNN [9]. There are also some pretrained models for this task [15], [16]. For example, Honnibal et al. [15] released a package tool called Spacy for NER, noun phrase chunking and other annotation tasks.

C. Concept Extraction

Concept Extraction is a recently proposed research task, which aims to find words or phrases describing a concept from massive texts [10]. Li et al. [10] first utilized many models to generate possible concepts and then designed a mechanism to evaluate the fitness of extracted concepts to original text.

To summarize, the above studies focus on their respective target phrases and cannot be directly transferred into technical phrase extraction. First, it is unsuitable to apply supervised methods as there are not enough labeled technical phrases from massive patent data. Besides, unsupervised approaches are often sensitive to the extraction target, so there are certain gaps between technical phrase extraction and existing methods.

III. Description of Technical Phrase

In this part, we will give a clear description of technical phrases in patents based on expert experience and statistics. Specially, we hire four experts to manually extract technical phrases from 100 patents in two domains, i.e., Electricity and Mechanic Engineering respectively. Each patent contains a multi-level structure, i.e., "Title", "Abstract" and "Claim", where "Title" and "Abstract" depict the topic and brief summary of a patent, and "Claim" is a more detailed and lengthy description of inventor's rights. From the extracted technical phrases in patents, we have some specific observations:

1) *Part of Speech*. Although the part of speech distribution of technical phrases shows various types, most of them are noun phrases and the percentage exceeds 90%.

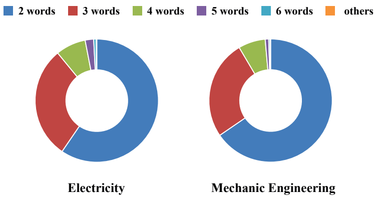
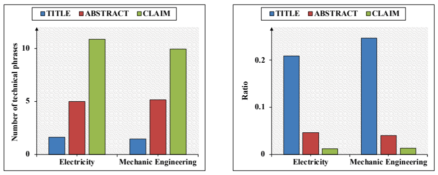


Fig. 1: Number of Words in Technical Phrases



(a) (b)

Fig. 2: (a) The average number of technical phrases in "Title", "Abstract" and "Claim". (b) The average ratio of the number of technical phrases to the number of words in different levels.

2) *Number of Words*. As shown in Fig. 1, the lengths of technical phrases are slightly different, but most of them are composed with 2 $\sim$ 4 words, sometimes reaching 5.
3) *Semantic Context*. In a patent document, there often exist similar technical phrases, such as "image encoding" and "image decoding". Naturally, these technical phrases will be relatively more similar in semantics. Besides, technical phrases ought to have a relatively independent technology meaning, and some phrases like "system architecture" also occur coupled with technical phrases frequently, but they are not our aim as there is no specific technical meaning.
4) *Local Occurrence*. On each level of a patent, technical phrases often appear more than once especially in long texts, which can be seen as a local occurrence. For example, across the extracted phrases from "Claim", over 70% of technical phrases appear at least twice in the text.
5) *Global Occurrence*. In a patent, one technical phrase tends to appear repeatedly across different levels. Their global occurrence in multi-level structures may provide some insights for aiding technical phrase recognition. To verify this point, a focused analysis is conducted in the following.

Fig. 2 (a) shows the average number of technical phrases across different levels. As we can see, the number of technical phrases grows rapidly from "Title" to "Claim" on both two datasets. Fig. 2 (b) shows the average ratio of the number of technical phrases to the number of words in different levels. From "Title" to "Claim", this ratio drops a lot, indicating that more and more non-technical phrases come out, and the difficulty to recognize technical phrases greatly increases.

Meanwhile, over 35% "Abstract"s contain at least one same technical phrase from "Title"s, and this percentage rises to 80% when it comes to "Claim"s and "Abstract"s. In other words, technical phrases from current level (e.g., "Title") may play a guiding role in the extraction of next level (e.g., "Abstract"), which can formulate a multi-level model architecture.

Moreover, existing patent classification systems can be an initial driving force for technical phrase recognition, for example, CPC Group (Cooperative Patent Classification Group), whose descriptions (e.g., multiplex communication, wireless

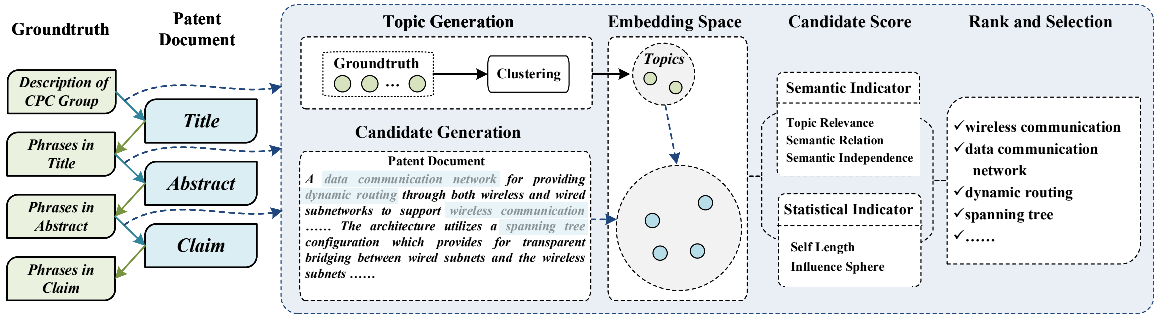


Fig. 3: The UMTPE Framework

communication networks) are highly relevant to technologies. Although their quantity is limited, we can still regard them as groundtruth examples to help the extraction in the first level (i.e., "Title") of patent documents.

IV. Problem Formulation

Considering the costliness of labeled data, in this paper, we aim to design an unsupervised method to extract technical phrases from patents. Specifically, based on the characteristics of technical phrases and multi-level structures of patent documents, we attempt to extract technical phrases level by level, where the extracted phrases in current level will be seen as groundtruth examples for guiding the next level, and CPC Group descriptions can be seen as the initial level.

In detail, for each level of a patent, technical phrase extraction is formulated as a generation and selection problem. That is to say, given the word sequence of a patent document $\boldsymbol{x} = (x_1, x_2, ..., x_n)$, we first build a candidate pool $\boldsymbol{Y} = \{y_i, i = 1, 2, ...\}$, where $y_i = (x_m, x_{m+1}, ..., x_n)$ is a possible technical phrase. Next, we design a score and rank mechanism to select final technical phrases from $\boldsymbol{Y}$. Finally, with technical phrases extracted from "Title", "Abstract" and "Claim", we can obtain all technical phrases for each patent.

V. The UMTPE Model

Based on the description of technical phrases in Section III, we develop an Unsupervised Multi-level Technical Phrase Extraction (UMTPE) model to recognize technical phrases from massive patent text. As shown in Fig. 3, UMTPE model deals with patent data level by level. And in each level of patents, UMTPE model contains five modules: Candidate Generation, Phrase Embedding, Topic Generation, Candidate Score, Candidate Rank and Selection.

A. Candidate Generation

In order to improve the completeness of extracted phrases, this part constructs a large-scale candidate phrase pool via several phrase extraction tools including Autophrase, DBpedia and Spacy, and then add one noun phrase extraction regulation.

1) Autophrase [12]: This model extract salient phrases based on quality estimation and occurrence identification.

2) DBpedia [16]: It is a tool for automatically annotating mentions of DBpedia resources in text.

3) Spacy [15]: We use the entity and noun phrase chunking part of Spacy to generate candidate phrases.

4) Noun Phrase Extraction [17]: As we mentioned in Section III, the majority of technical phrases are noun phrases. In order to avoid missing some candidates, we extract more noun phrases to complement this pool using grammar tagging.

After that, we filter out all single words and merge phrases from four methods after removing duplications.

B. Phrase Embedding

The UMTPE methodology to some extent relies on the underlying embeddings. Considering the overlapping problem of candidate phrases in a sentence, we first train unigram embeddings using skip-gram model and then average their dense vectors to obtain vectors for multi-word phrases.

C. Topic Generation

We first map the content in CPC Group to embedding space, and cluster them to a few centroids, which aims to find several topics of technical phrases. Then these topic centroids will be utilized to guide the phrase extraction in "Title". After that, we will select technical phrases with high confidence from extracted results in "Title" to do the same things to "Abstract", which will form a multi-level structure in Fig. 3.

Rather than focusing on a certain patent, the topic generation concerns all groundtruth examples or highly confident technical phrases from a certain level across the dataset. This design can overcome the effect of few bad cases and improve the robustness. As for the choice of clustering method, we use a hierarchical clustering method called HDBSCAN [18].

D. Candidate Score

In this part, we construct a graph composed of candidate phrases and score them from semantic and statistical angles.

1) Graph construction: The inner-connections of candidate phrases play an important role in mining candidate's differences and relations, which are crucial for technical phrase discrimination. Considering that, we would like to establish a candidate phrase graph for a certain patent document. In the graph, nodes are all candidate phrases, and edge's weight between arbitrary two nodes is their cosine similarity.

2) Semantic Measurement Indicators: Upon the base of the graph and topic centroids, we design three measurement indicators to score every candidate from the semantic perspective.

- **Topic Relevance** measures the relevant degree between candidate phrases and existing topics from last level. High

Topic Relevance means this candidate is more associated with a specific technology topic. We define it as the largest cosine similarity between the candidate and topic centroids:

$$Topic\_relevance_i = \max_k \cos(\theta_i, Topic_k). \quad (1)$$

- **Semantic Relation** measures the link ability of technical phrases. In general, similar technologies tend to appear at the same context, such as the closely associated technical phrases “image encoding” and “image decoding”. To better quantify Semantic Relation, we first cut edges in the graph whose weight is smaller than a threshold T and define the indicator as the normalized degree of the candidate node:

$$Semantic\_relation_i = \frac{\sum_{j \neq i} \mathbb{I}(\cos(\theta_i, \theta_j) \geq T)}{\sum_{j \neq i} \mathbb{I}(1)}. \quad (2)$$

- **Semantic Independence** focuses on the independence of technical phrases in the semantic embedding space. As we mentioned in Section III, technical phrases also need a relatively independent meaning. We define this indicator as the smallest cosine distance with other nodes in the graph:

$$Semantic\_independence_i = \min_{j \neq i}(1 - \cos(\theta_i, \theta_j)). \quad (3)$$

3) Statistical Measurement Indicators: With the observations in Section III, we also design two intuitive statistical measurement indicators.

- **Self Length** counts the number of words in the candidate phrase. From the analysis in Fig. 1, most technical phrases are composed of 2∼4 words, and sometimes the number reaches 5. According this finding, we define Self Length as:

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

where $len(\theta_i)$ represents the word number of the candidate.

- **Influence Sphere** measures the influence scope of a candidate phrase. On each level of a patent, technical phrases often appear in more than one sentence as they are crucial for relating different parts in the paragraph. From this perspective, we define this indicator as the number of sentences including the candidate phrase in current document:

$$Influence\_sphere_i = \sum_k \mathbb{I}(\theta_i \in sentence_k). \quad (5)$$

The statistical measurement indicators mentioned above are designed more intuitively, while semantic measurement indicators focus on the inner-connections of candidate phrases. Based on these measurement indicators, we can comprehensively evaluate every candidate in a graph, and the normalized sum of these scores will be set as the weight of nodes.

E. Candidate Rank and Selection

1) Candidate Rank: In this part, we conduct NE-rank algorithm [14] on the candidate graph. NE-rank is an improved ranking algorithm based on pagerank and textrank, which ensures the communication between nodes and allows us to comprehensively evaluate every node in the graph. Then we will get a ranking list of candidate phrases for each document.

2) Candidate Selection: With the results of NE-rank, we select top-K as technical phrases, while the candidates of high confidence (top-1) will be put in a new groundtruth set to be sent to next level. Considering the contents in different documents vary a lot, we set K according to the number of sentences in the document (N_{sen}). From the labeled data, we calculate the statistical relation between K and N_{sen}:

$$\frac{K}{N_{sen}} \approx \begin{cases} 1 \sim 2 & Title, \\ 2 & Abstract, \\ 1 & Claim. \end{cases} \quad (6)$$

Based on this observation, we set $K = 2N_{sen}$ for patent “Title” and “Abstract”, $K = N_{sen}$ for patent “Claim”.

VI. Experiment

A. Experimental Setup

1) Datasets: Experiments are performed on USPTO[2] patent data in two domains, i.e., Electricity and Mechanical Engineering. The former is related to electric field, and the latter concerns mechanical engineering, lighting, heating, etc. Specially, we randomly sample 84k and 11k pieces of patent data in Electricity and Mechanical engineering datasets respectively.

2) Implementation Details: In this part, we describe the implementation details of UMTPE model. We run all experiments on one Tesla K80 GPU and 16 Intel CPUs.

- *Topic Generation Hyperparameters*: In HDBSCAN algorithm, We set the *min_cluster_size* as 3 for the level of CPC Group, while 100 for others (i.e., “Title” and “Abstract”).
- *Candidate Score Hyperparameters*: We set $T = 0.5$ in Eq. (2) of Semantic Relation measurement indicator.

3) Baselines: We compare with a wide range of state-of-the-art approaches, as described below[3]:

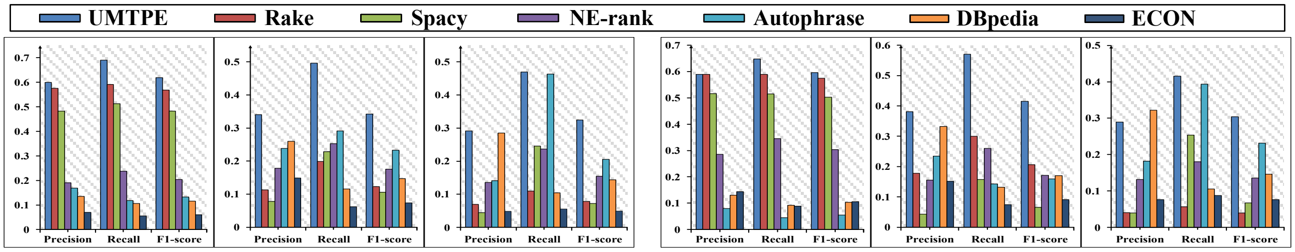
- **Autophrase** [12], **DBpedia** [16] and **Spacy** [15]. These three phrase extraction models in the Candidate Generation part are certainly in our baseline group.
- **Rake** [13] uses graph-based importance measurement and adjacency relations to extract key phrases.
- **NE-rank** [14] is utilized for phrase candidate ranking in our model. Actually, in the original paper, NE-rank can also be utilized directly to extract phrase from documents.
- **ECON** [10] proposes to extract concept word/phrase based on embedding and probability theory.

B. Result

1) Overall Performance Evaluation: In this part, we evaluate the overall performances on 100 labeled patents using three widely-used metrics in various applications [8], [19], [20], i.e., *Precision*, *Recall* and *F1-score*. We first get the prototype of every word in reference and predicted phrases, and then calculate the results of each level in patents (“Title”, “Abstract”, “Claim”). In order to evaluate results comprehensively, we average the results in three levels according to the radio of

[2] https://www.uspto.gov

[3] Among these baselines, some models like ECON will extract both phrases and words. For fair comparison, we filter out all single words. For baselines giving phrases in a certain ranked order, we also select top-K phrases.



(a) Title (b) Abstract (c) Claim (d) Title (e) Abstract (f) Claim

(1) Electricity (2) Mechanical Engineering

Fig. 4: Overall performance Evaluation on Electricity and Mechanical Engineering Dataset

TABLE II: Overall Performance Evaluation (%)

	Electricity			Mechanical Engineering		
Method	Precision	Recall	F1-score	Precision	Recall	F1-score
UMTPE	**41.04**	**55.21**	**42.85**	**41.53**	**53.89**	**43.37**
Rake	25.25	30.00	25.64	26.93	31.54	27.33
Spacy	20.17	32.89	22.01	19.99	30.86	21.16
NE-rank	16.80	24.22	17.78	19.08	26.20	20.34
Autophrase	18.25	29.11	19.06	16.53	19.37	14.87
DBpedia	22.66	10.91	13.56	26.16	10.96	13.99
ECON	8.92	5.74	6.07	12.36	8.34	9.06

TABLE III: Representation Evaluation (%)

	Electricity		Mechanical Engineering	
Method	Abstract	Claim	Abstract	Claim
UMTPE	**55.16**	**50.38**	**55.24**	**44.57**
Rake	**64.78**	**48.89**	**64.64**	**41.15**
Spacy	45.73	29.56	40.63	27.16
NE-rank	44.10	34.75	41.29	27.97
Autophrase	43.22	27.33	28.36	27.97
DBpedia	20.83	11.45	19.47	8.65
ECON	15.29	15.49	20.47	17.10

1:1:1. The final results are listed in Table II, and the results in each level are shown in Fig. 4.

As we can see in Table II, UMTPE model outperforms all baselines in all metrics, which proves the effectiveness of multi-level architecture coupled with semantic and statistical measurement indicators. Then across different levels, we find some interesting phenomena: 1) the differences between the UMTPE model and baselines in "Title" are less obvious than results in other levels. That is because the majority of "Title"s are often short texts with only one sentence, which reduces the extracting difficulty extensively. 2) When it comes to long text, i.e., "Abstract" and "Claim", performances of these baselines drop a lot. Moreover, the performance of Autophrase steadily grows from "Title" to "Claim", because its extraction basis greatly relies on the fluency of possible phrases, which is more suitable for longer documents. All in all, despite the advantages of different models, our UMTPE model achieves best in task of technical phrase extraction.

2) Representation Evaluation: In order to supplement traditional evaluation metrics, we propose a new metric called Information Retrieval Efficiency (IRE) to evaluate the predicted technical phrases from the angle of representation ability. As we discussed in Section I, the combination of technical phrases can make a technology portrait for patents, which carries essential and distinctive technical information of patents. From this perspective, technical phrases are expected to have more powerful representation ability than general phrases, so IR task on patent documents could verify extraction results effectively.

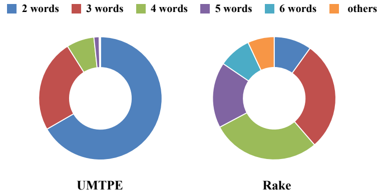


Fig. 5: Word Number of Phrases from UMTPE and Rake

Therefore, we conduct an IR task in size of 1,000 patent documents including those 100 patents with labeled technical phrases. For every predicted phrase in a document, we use it as a query to rank all documents according to the matching degree[4]. If the document where this phrase comes from is in the top-10 documents set, we score the phrase as 1. Otherwise the score will be 0. Then, we compute the score of this document by averaging scores of all extracted phrases. On the basis of this score, we also design a penalty factor PF to avoid the effect of fewer extraction phrases[5].

$$PF = \begin{cases} 1 & r \leq p, \\ e^{1-r/p} & r>p, \end{cases} \tag{7}$$

where r is the number of reference technical phrases in the document and p is the number of phrases extracted by the model. With PF, we can revise the score for every document:

$$score_{revise} = PF \cdot score. \tag{8}$$

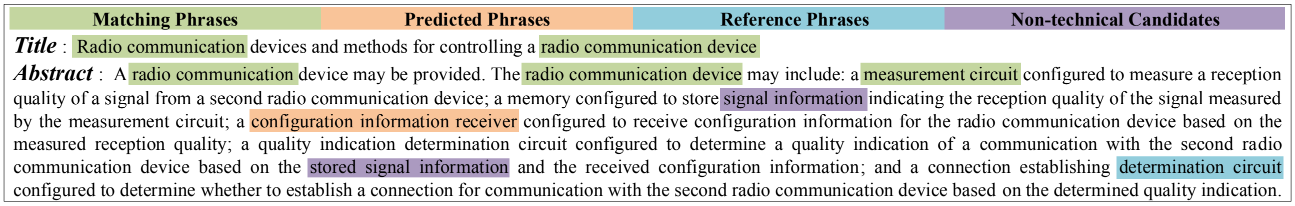
At last, we average the revised score of these 100 labeled documents to get the final value of IRE. The results on "Abstract" and "Claim" are listed in Table III[6].

From the results in Table III, we can find that the phrases extracted by UMTPE and Rake both have excellent representation ability and is far ahead of other models. However, the characteristics of extracted phrases form UMTPE and Rake seems quite different. Fig. 5 shows the number of words in phrases extracted by UMTPE and Rake. As we can see, phrases extracted by UMTPE are consistent with reference phrases in Fig. 1, while Rake tends to extract phrases with more words. In general, it is a natural thing that more words indicate more information and thus better performance for IR tasks. Therefore, it is explicable that high performance of Rake

[4] We use LSI (Latent Semantic Indexing) model to do this task.

[5] This circumstance means that if a model only extracts one or two high-quality phrases from a document containing ten technical phrases, the score on this document still tends to be very high.

[6] We have not shown this evaluation on "Title" as most of them are composed with one sentence, which is quite easy for IR task and unsuitable for evaluating the performances of technical phrase extraction results.



Matching Phrases | Predicted Phrases | Reference Phrases | Non-technical Candidates

Title : Radio communication devices and methods for controlling a radio communication device

Abstract : 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. 6: Case Study

tends to benefit from the extracted lengthy phrases. On the contrast, UMTPE can not only extract technical phrases in line with the actual situation but also outperform Rake on "Claim", which indicates that phrases extracted by UMTPE show a great advantage in representing technical information.

C. Case Study

In this subsection, we conduct a case study to further explain the results of UMTPE model. For better visualization, we show the extraction results from one "Title" and "Abstract" in Fig. 6. In this case, "matching phrase" means the same phrases extracted by UMTPE model and human labeling, while "reference phrase" and "predicted phrase" represent the phrase only extracted by human labeling or UMTPE respectively. From this illustration, we can find that UMTPE can accurately recognize technical phrases in both "Title" and "Abstract", such as "radio communication". And the technical phrases in "Title" are included by technical phrases in "Abstract", which also verifies the effectiveness of the multi-level design.

VII. Conclusion and Future Work

In this paper, we explored a motivated direction for technical phrase extraction in patent data. Specifically, we first gave a clear and detailed description about technical phrases in patents. Then, combining characteristics of technical phrases and multi-level structures of patent data, we developed an Unsupervised Multi-level Technical Phrase Extraction (UMTPE) model, which could recognize technical phrases and was free of expensive human labeling. After that, we designed a novel metric called Information Retrieval Efficiency (IRE) to evaluate extracted phrases from the perspective of representation ability, which could supplement traditional evaluation metrics. Finally, we evaluated UMTPE framework on real-world patent data and experimental results clearly proved its effectiveness. We hope this work could lead to more future studies.

VIII. Acknowledgements

This research was partially supported by grants from the National Key Research and Development Program of China (No. 2016YFB1000904), the National Natural Science Foundation of China (No. U1605251, 71802068) and the provincial projects on quality engineering for colleges and universities in Anhui Province (No. 2020zdxsjg400).

References

[1] Longhui Zhang, Lei Li, Tao Li, and Qi Zhang. Patentline: analyzing technology evolution on multi-view patent graphs. In *SIGIR*, 2014.

[2] Yan Liu, Pei-yun Hseuh, Rick Lawrence, Steve Meliksetian, Claudia Perlich, and Ro Veen. Latent graphical models for quantifying and predicting patent quality. In *In KDD'11*. Citeseer, 2011.

[3] Han Wu, Kun Zhang, Guangyi Lv, Qi Liu, Runlong Yu, Weihao Zhao, Enhong Chen, and Jianhui Ma. Deep technology tracing for high-tech companies. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1396–1401. IEEE, 2019.

[4] Qi Liu, Han Wu, Yuyang Ye, Hongke Zhao, Chuanren Liu, and Dongfang Du. Patent litigation prediction: A convolutional tensor factorization approach. In *IJCAI*, pages 5052–5059, 2018.

[5] Longhui Zhang, Lei Li, and Tao Li. Patent mining: a survey. *ACM Sigkdd Explorations Newsletter*, 16(2):1–19, 2015.

[6] Yuen-Hsien Tseng, Chi-Jen Lin, and Yu-I Lin. Text mining techniques for patent analysis. *Information processing & management*, 43(5):1216–1247, 2007.

[7] Hongjie Lin, Hao Wang, Dongfang Du, Han Wu, Biao Chang, and Enhong Chen. Patent quality valuation with deep learning models. In *International Conference on Database Systems for Advanced Applications*, pages 474–490. Springer, 2018.

[8] Kazi Saidul Hasan and Vincent Ng. 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)*, pages 1262–1273, 2014.

[9] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[10] Keqian Li, Hanwen Zha, Yu Su, and Xifeng Yan. Concept mining via embedding. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 267–276. IEEE, 2018.

[11] Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. Deep keyphrase generation. *arXiv preprint arXiv:1704.06879*, 2017.

[12] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837, 2018.

[13] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20, 2010.

[14] Abdelghani Bellaachia and Mohammed Al-Dhelaan. 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, pages 372–379. IEEE, 2012.

[15] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1), 2017.

[16] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, pages 121–124, 2013.

[17] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[18] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 2017.

[19] Qi Liu, Yong Ge, Zhongmou Li, Enhong Chen, and Hui Xiong. Personalized travel package recommendation. In *2011 IEEE 11th International Conference on Data Mining*, pages 407–416. IEEE, 2011.

[20] Hao Wang, Enhong Chen, Qi Liu, Tong Xu, Dongfang Du, Wen Su, and Xiaopeng Zhang. A united approach to learning sparse attributed network embedding. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 557–566. IEEE, 2018.