Technology Effect Phrase Extraction in Chinese Patent Abstracts

Dacheng Liu, Zhiyong Peng, Bin Liu*, Xu Chen, and Yuqi Guo

School of Computer, Wuhan University 129 Luoyu Road Wuhan, China
{dacheng.liu,peng,binliu,chenxu,yuqi.guo}@whu.edu.cn

Abstract. Patents are the greatest source of technical information in the world. High-efficient patent mining technologies are of great help to technical innovations and protection of intellectual property right. The extraction of technology effect clauses or phrases is an important research area in patent mining. Due to the specialty and uniqueness of patent data, traditional keyword extraction algorithms cannot properly apply to the extraction of technology effect phrases, leaving it dependent on high-cost manual processing. We propose a semi-automatic method based on partitioning corpus to extract technology effect phrases in Chinese patent abstracts. Experiments show that this method achieves satisfying precision and recall while involving little human labor.

Keywords: Patent mining, information retrieval, technology effect clause, technology effect phrase, partitioning corpus.

1 Introduction

Patents are the greatest source of technical information in the world. Statistics show that patents contain 90%-95% of the information in today's scientific and technological innovations. For any enterprise, patents are the key technical information that must be made public. By analyzing patent data, people can obtain valuable information to avoid wasting money on redundant research and to prevent property rights violations. Before any technical innovations, researchers must thoroughly learn about the existing patents in the targeted domain[1].

The extraction of technology effect clauses (**TEC**) is an important research area in patent mining. Patent technology effect matrix analysis has attracted much attention these years for its ability to reveal important hidden information in patent data[2]. TEC's point out technologies used in patents and effects they achieve. They are good raw materials for in-depth analysis of the patent data in a given domain. The extraction of TEC's with high precision and recall is the foundation of this technology. Patent abstracts contain most useful information in patent data[2], and there are many tools to help patent analyzers to download patent abstracts in bulk. Therefore, we focus our extraction mainly on patent abstracts. Note that patent data

* Corresponding author.

L. Chen et al. (Eds.): APWeb 2014, LNCS 8709, pp. 141–152, 2014.

of different domains differ much in content. To guarantee high recall, we execute the extraction in one domain each time.

At present, TEC extraction depends on technology effect phrases (**TEP**). In the domain of Chinese patent mining, few algorithms are able to effectively extract TEP's without much human assistance. To ensure high precision and recall, patent mining algorithms invariably rely on manual extraction, rather than traditional keyword extraction algorithms. Most of these algorithms consider word frequency as a decisive component in determining keywords. However, the uniqueness of patents makes word frequency in patent abstracts fairly low. Therefore, we propose a novel method that is independent of word frequency and that requires little human involvement.

This paper is organized as follows: Section 2 discusses related works. Section 3 presents our algorithm and its improvements. Section 4 shows our experiment results. Section 5 summarizes this paper and discusses future work.

2 Related Works

2.1 Technology Effect Annotation in Patent Data

The number of patents has greatly increased in recent decades, and patent retrieval technologies are also developing fast. Current researches mainly focus on the annotation of patent function, technology, and composition parts[3]. In English and Japanese patent processing, large amount of manually annotated patent data have provided good corpora for some machine learning algorithms, such as the rule-based method by Peter Parapaics[5] and many supervised training methods proposed on NTCIR Workshops[4]. Using these algorithms, people have achieved remarkable results in technology effect annotation. Due to the lack of annotated Chinese patent data, Xu Chen[3] proposes a semi-supervised co-training algorithm to annotate TEC's in Chinese patent abstracts without using a large corpus. By several times of iteration, the algorithm achieves an F-Measure over 80%. However, in each iteration, manual processing is needed to extract TEP's from newly identified TEC's. Too much human labor prevents this method from practical use.

2.2 Keyword Extraction and Technology Effect Phrase Extraction

Many state-of-the-art algorithms are available for extracting keywords from general text. Y. Matsuo[6] et al. propose a domain-independent keyword extraction algorithm that needs no corpus. The algorithm measures the distribution of the co-occurrence of frequent terms and other terms to determine the importance of words. It achieves good performance in experiments on general text, but the importance of TEP's is determined by their semantic meanings, rather than how important they seem according to the rules in [6]. Besides, traditional keyword extraction algorithms are dependent on word frequency, but TEP's are hard to distinguish from other phrases in this sense. Ni W.[7] et al. propose an under-sampling approach to extract imbalanced key phrases in general text. The algorithm is based on a training set represented by two views of the samples, i.e. the morphology view and occurrence view of the (non-)key phrases. In TEP extraction, however, the occurrence of TEP's are hard to distinguish from other phrases. Besides,

we have tried various approaches to identify TEP's using their morphologies by an LVQ neural network as a classifier but none of them works well. The difficulty lies in isola ting the morphologies belonging only to TEP's. Ying Chen[2] et al. propose a method based on patent structure-grammar-keyword feature to identify the TEP's in patent data. Because DII (Derwent Innovations Index) is an essential part of this algorithm, and the construction of DII demands much processing by patent experts, the algorithm is not successful in reducing human labor.

2.3 Parsing Chinese Patent Data

Roger Levy[8] et al. carefully investigate the errors that Stanford Parser[9] makes in parsing general Chinese text and provide some simple and targeted improvements for each case. These improvements enable the parser to achieve high accuracy in parsing general Chinese text. Unfortunately, after parsing some Chinese patent abstracts with Stanford Parser, we find it almost impossible to distinguish the subtrees containing a TEP from other subtrees. We even notice that a few subtrees containing a TEP are identical with other subtrees in both tree structure and part-of-speech tagging (See Figure 1), making it hard to extract TEP's by the parsing results. The Chinese Academy of Sciences develops a word-segmenting and part-of-speech tagging software ICTCLAS [10]. The software does well in general Chinese text such as news, magazine articles, and literature works. Though it is now the best Chinese text tokenizer, our experiment shows that it often segments terminologies in patent abstracts into separated parts(See Figure 2). It also wrongly tags a few words in almost every patent abstract.

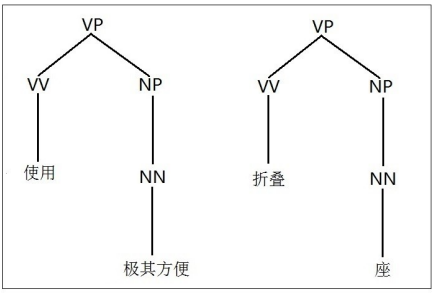

Fig. 1. Identical parse trees are generated for both a TEP (“convenient + usage”) and an unexpected phrase (“fold + foundation”)

本/rz 实用/a 新型/b 在/p 支撑/v 架/qv 的/ude1 前/f 、/wn 后/f 盖/v 之间/f 设置/v 有/vyou 铁/n 制/k 加强板/n ，/wd 提高/v 了/ule 支撑/v 架/qv 的/ude1 稳固/a 性/ng ，/wd 避免/v 了/ule 固定座/n 在/p 移动/vn 或/c 转动/v 回转架/n 时/ng 产生/v 晃动/v 。/wj

Fig. 2. In a TEP “提高+稳固性” (“enhance + stability”), “稳固性”(stability) is wrongly segmented into “稳固”(stable) and “性”(sex) with wrong word tags

3 Proposed Method

3.1 Definitions

For convenience, we give definitions to some frequently used terms in this paper.

Definition 1. Technology Effect Clause (TEC) A technology effect clause (TEC) is a clause describing the technology and effect of a patent.

Definition 2. Technology Effect Phrase (TEP) A technology effect phrase (TEP) is a phrase that seldom appears in a non-TEC. Once it appears, the clause containing the phrase must be a TEC.

There are mainly two forms of TEP's: "verb + noun" and "adjective + noun". We unify the two forms by "Value + Attribute". E.g. "增加 + 产量" (increase + production) and "高 + 可靠性" (high + reliability) are effect phrases, and "使用 + 环保材料" (uses + green material) is a technology phrase.

Definition 3. Value and Attribute Value is the verb or adjective part in the above two common forms of TEP, and Attribute is the noun part.

Definition 4. Domain-Independent Corpus (DIC) A Domain-Independent Corpus (DIC) is a pre-constructed corpus consisting of Values and Attributes shared by most patent domains.

Definition 5. Domain-Dependent Corpus (DDC) A Domain-Dependent Corpus (DDC) is a corpus constructed using Values and Attributes in the targeted domain.

3.2 The Basic Idea of the Algorithm

We have shown in Section 2.3 that grammar-structure-based methods fail in TEP extraction. On the other hand, manual extraction achieves high recall and precision because patent analyzers know what phrases could be TEP's, yet they hardly parse the text to see which part of the parse tree may correspond to a TEP. Therefore, a corpus is needed for an effective and automatic extracting method. However, constructing a corpus of TEP's directly is impractical for the following two problems: (We denote the set of patent abstracts by **A**)

Problem 1. Word redundancy is high in the TEP corpus. E.g. the Value "提高" (improve) can be matched with many Attributes such as "质量" (quality), "产量" (production), and "效率" (efficiency). Including all these phrases into the corpus seems unnecessary.

Problem 2. If all TEP's in **A** have been manually extracted to construct the corpus, then the corpus is just the result of manual extraction. It is meaningless to use this corpus to extract the patent abstracts.

For Problem 1, we partition the corpus into a "Value corpus" and a "Attribute corpus". The algorithm first performs a Cartesian product of the Values and the Attributes. We denote this result set by **R**. Then for every patent abstract $\mathbf{P} \in \mathbf{A}$, the algorithm scans for every phrase $\mathbf{e_i} \in \mathbf{R}$ to see if it appears in **P**. If so, we consider $\mathbf{e_i}$ as a TEP candidate.

We address Problem 2 based on a fact we observe in our experiments: the content of patent data in a certain domain is highly domain-dependent. Let **B** be a subset of **A**

formed by random sampling. The number of elements in **B** is far less than that of **A**. This observation is presented as follow:

Observation 1. Let **A** be a set of patent abstracts. Let **B** be a subset of **A** where M<|**B**|<|**A**|. M is a small positive integer. The technologies and effects mentioned in **B** are also frequently mentioned in the set **A** – **B**, and those technologies and effects in **A** – **B** but not in **B** are very rare.

This observation indicates that we can construct a corpus C_B using only the patent abstracts in **B**. The key is that corpus C_B is able to generate new TEP's. There could be many different expressions for one technology or effect. E.g. let there be "提高+效率" (improve + efficiency) and "增加+稳定性" (increase + stability) in **B**, C_B has the potential to generate "提高+稳定性" (improve + stability) and "增加+效率" (increase + efficiency), which are quite likely to appear in set **A** – **B**.

Further observations show that there are domain-independent Values and Attributes. Values such as "提高"(improve) and "防止"(prevent) and Attributes such as "成本"(cost) and "质量"(quality) are likely to appear in most patent abstracts. Meanwhile, Values like "制动"(brake) and "燃烧"(burn) and Attributes like "油耗"(oil consumption) and "机械效率"(mechanical efficiency) are dependent to the domain of diesel engine. To further reduce human labor in corpus construction, we partition both the corpora of Values and Attributes into domain-independent corpus (**DIC**, Definition 4) and domain-dependent corpus (**DDC**, Definition 5). We have constructed DIC for both Values and Attributes according to our observation of large quantities of patent abstracts. When a patent analyzer wants to extract patent abstracts in a new domain, he only needs to manually extract domain-dependent Values and Attributes (especially Attributes) from a small number of patent abstracts. Then the algorithm will incorporates the DIC and DDC and eliminates duplicate elements during extraction process. We show the details of the partitioning in Figure 3.

3.3 Improving Precision and Recall

Though recall of the method in Section 3.2 is acceptable, its precision is only around 50%. This is due to some certain common Chinese language grammar structures. We accordingly propose the following rules to improve precision:

Rule 1. If the character "了" appears in a clause and there is a Value identified right before this "了", then this Value is considered to be the only Value of this clause.

*The necessity of this rule is substantiated by the abundance of TEP's presented in the form "Value+*了*+Attribute". E.g. in clause "在生产过程中减小了成本" (decrease cost in production) we extract "减小" (decrease) as the only Value for the Attribute "成本" (cost), ignoring the verb "生产" (produce).*

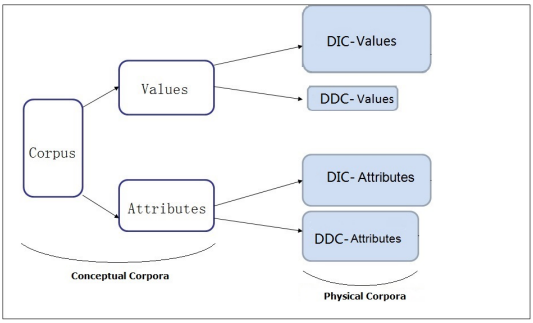

Fig. 3. Corpus partition. The three hollow boxes on the left represent the conceptual corpora in our analysis. The four solid boxes on the right are the physically existent corpora used in our algorithm. The sizes of the four solid corpora reflect their actual sizes in our experiment. The pre-constructed DIC-Values plays an important role in the extraction.

Rule 2. If a Value matches several Attributes in a clause, and none of these Attributes is a substring of another, we check if there is an Attribute right behind a character "的". If so, we match the Value with only this Attribute.

*This rule is needed for a specific yet common case where an Attribute is embellished by another word before it. E.g. in clause "*增加单位体积燃料的里程数 *"(increase the mileage per unit fuel), "*里程数 *"(mileage) rather than "*燃料 *"(fuel) is the correct Attribute.*

Rule 3. If a Value matches several Attributes in a clause, and some of these Attributes are substrings of others, only the longest Attribute should be matched.

This rule is designed to eliminate obvious duplications. Still using the example of "增加单位体积燃料的里程数 "(*increase the mileage per unit fuel), if* "单位体积燃料的里程数 "(*mileage per unit fuel) appears so frequently that it is included in the DDC of Attribute, it is unnecessary to consider "*里程数 *"(mileage) as an Attribute.*

Summarizing Section 3.2 and 3.3, we now give a formal description of our algorithm in Table 1. A flowchart of the entire method is shown in Figure 4.

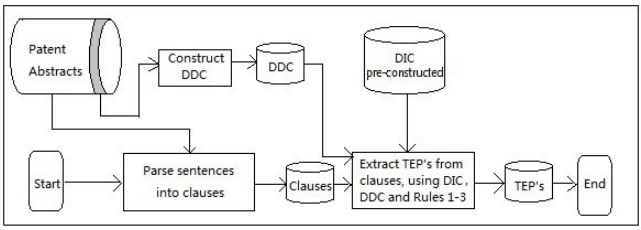

Fig. 4. The flowchart of the entire TPE extraction method

Table 1. Algorithm for Extracting TEP's from Chinese Patent Abstracts

Algorithm: Technology Effect Phrases Extraction Based on Partitioning Corpus **Input**: DIC (Domain Independent Corpus), DDC (Domain Dependent Corpus), m patent abstracts

Output: All technonogy effect phrases in the m patent abstracts

1. Load DIC and DDC, combine them with duplicate elimination.
   Put all Values in **Values**$[]_{n1}$, all Attributes in **Attributes**$[]_{n2}$;
2. Load patent abstract $\mathbf{A_i}$, segment it by '，', '。' and '；', save the result in **Array**$[]_p$;
3. For each sub-sentence $\mathbf{S_j}$ in **Array**$[]_p$
4. Apply **Rule 1** and **Rule 2** (in order) to $\mathbf{S_j}$;
5. If either rule takes effect, then output the phrase(s) extracted by the rule, **GOTO** Step 3 ;
6. Apply **Rule 3** to $\mathbf{S_j}$;
7. For each Value $\mathbf{V_x}$ in **Values**$[]_{n1}$ and each Attribute $\mathbf{A_y}$ in **Attributes**$[]_{n2}$
8. IF $\mathbf{V_x}$ and $\mathbf{A_y}$ are found in $\mathbf{S_j}$, then output "$\mathbf{V_x} + \mathbf{A_y}$" as a TEP, **GOTO** Step 3 ;
9. End For
10. End For
11. **GOTO** Step 2 until all patent abstracts are exhausted;

The main step of the algorithm may seem straightforward. However, we have made great efforts trying to come up with algorithms based on supervised learning, referring to methods mentioned in Section 2. All these approaches fail to properly adapt to the specific job of TEP extraction. Because they either show disappointing recall and precision or require too much manual processing. Compared with these methods, our method substantially reduces manual processing while achieving relatively high precision and recall. We also compare our method with two standard keyword extraction algorithms in Section 4.4. Our method achieves obviously higher recall and precision.

3.4 The Number of Manually Extracted Patent Abstracts

In Observation 1 a lower bound M is given for the number of manually extracted patent abstracts. It should be noted that M is not a fixed portion of the total patent abstract number. Suppose there are 100 patent abstracts in a domain, a patent analyzer may need to manually extract 20 abstracts to cover the domain-dependent Values and Attributes. But when the number rises to 1000, only about 60 abstracts should be pre-extracted. Since all these patents belong to one domain, the rest 940 patents are all focusing on what the 60 patents are doing. Therefore, the more patents in a domain, the less the portion for manual extraction, and the more cost-effective the method. We give the following function as a reference:

$$y = \lfloor 19ln(x) \rfloor - 66, (x \geq 100, x \in N) \quad (1)$$

Here y is the number of patent abstracts to manually extract, and x is the number of all patent abstracts in a domain. We use logarithm because we find the increment rate of y reduces significantly as x grows. The parameters in the function are determined using

Least Square Fit according to our experiment results. Further details will be given in Section 4.3.

4 Experiments

We download over 3000 patent abstracts from the website of State Intellectual Property Office of China. These patents are distributed in 7 domains, each consists of 102, 160, 263, 354, 459, 648, 1042 abstracts respectively. We first make a comparison between precisions of our method as Rule 1 to 3 are included to justify these rules. Then we show how precision and recall change as the number of manually extracted abstracts grows, and draw some interesting conclusions. Then we provide details on how the parameters in formula (1) are determined. Finally we compare our method with two standard keyword extraction algorithms.

4.1 Precision Increment by the Rules

We compare the precision before and after the introduction of each rule in Section 3.3 on the 7 domains respectively. We manually extract 20% abstracts in the first 4 domains and 10% abstracts in the following 3 domains[1]. The results are shown in Figure 5. A significant increase in precision occurs after each rule comes into play. We do not plot changes in recalls in Figure 5 because their decreases are negligible, and these rules are relatively conservative in this sense.

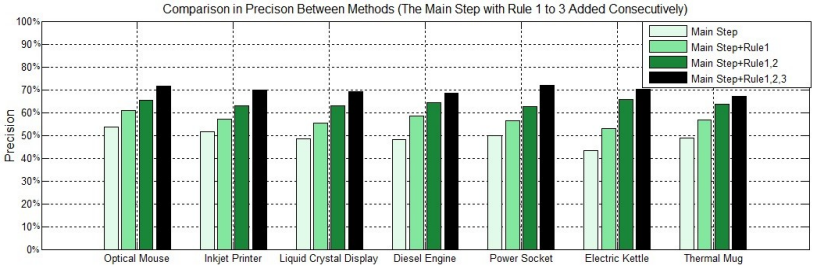

Fig. 5. Comparison in precision between methods with and without the rules

4.2 Precision and Recall

To test the overall precision and recall of our method, we run the algorithm on the 7 domains of abstracts respectively. For each domain, we start from zero manual extraction, then each time we manually extract a new 3% abstracts and add the Values and Attributes into DDC. We measure the precision and recall of TEP extraction based on Definition 2. Results are presented in Table 2 and Figure 6.

In Table 2, the underlined data represent the highest F-Measure achieved in that domain while precision is over 70%. The algorithm achieves F-Measures over 75% in all 7 domains, with recalls over 80% and precisions over 70%.

[1] Here the percentages of manual extraction are decided a little arbitrarily because we only intend to see if the rules work. To achieve the best F-Measure, one needs to follow the formula in Section 3.4.

Table 2. Changes in precision and recall as more abstracts are manually extracted. The first row is the domain name, including OM (Optical Mouse), IP (Inkjet Printer), LCD (Liquid Crystal Display), DE (Diesel Engine), PS (Power Socket), EK (Electric Kettle), and TM (Thermal Mug). The second row is the corresponding number of patents in each domain. In the first column, "M.E." stands for "Manually Extracted". In the second column, "P", "R", and "F" are "Precision", "Recall", and "F-Measure" respectively. A few cells are given by a "*", because the F-Measures have reached their peak before them, and testing them requires manually extracting over 100 abstracts.

Patent Domain		OM	IP	LCD	DE	PS	EK	TM
Num. of Patents		102	160	263	354	459	648	1042
0% M.E.	P%	67.0	68.1	65.3	66.7	64.5	68.2	66.4
	R%	**15.1**	**18.2**	**13.4**	**12.0**	**14.9**	**16.6**	**17.8**
	F%	24.6	28.7	22.2	20.3	24.2	26.7	28.1
3% M.E.	P%	69.1	73.1	76.7	75.8	77.4	76.3	73.2
	R%	**34.5**	**37.6**	**40.0**	**45.5**	**51.7**	**59.7**	**72.1**
	F%	46.0	49.7	52.6	56.9	62.0	67.0	72.6
6% M.E.	P%	75.9	74.4	74.2	72.1	73.8	73.8	71.3
	R%	**51.8**	**55.3**	**58.3**	**68.1**	**72.5**	**74.7**	**82.0**
	F%	61.6	63.4	65.3	70.0	73.1	74.2	76.3
9% M.E.	P%	73.7	73.6	72.6	70.4	71.2	71.4	67.9
	R%	**60.3**	**69.6**	**73.5**	**78.3**	**78.0**	**81.5**	**83.1**
	F%	66.3	71.5	73.0	74.1	74.4	76.1	74.7
12% M.E.	P%	72.0	71.9	71.5	70.5	70.3	69.0	
	R%	**68.8**	**77.1**	**80.1**	**81.9**	**82.8**	**82.8**	*
	F%	70.4	74.4	75.6	75.8	76.0	75.3	
15% M.E.	P%	72.5	71.0	71.2	69.6	69.2	68.8	
	R%	**75.4**	**79.5**	**83.6**	**82.5**	**83.0**	**83.4**	*
	F%	73.9	75.0	76.9	75.5	75.5	75.4	
18% M.E.	P%	71.9	70.2	70.3	68.5	68.3		
	R%	**78.9**	**81.3**	**83.9**	**83.0**	**83.1**	*	*
	F%	75.2	75.3	76.5	75.1	75.0		
21% M.E.	P%	71.4	69.9	68.9	68.6	67.0		
	R%	**80.6**	**81.8**	**84.1**	**83.2**	**83.5**	*	*
	F%	75.7	75.4	75.7	75.2	74.3		

Though precisions are obviously lower than full manual processing, patent retrieval is recall-oriented[12], and our recalls approximate manual performance. When a human is employed to do the extraction, things are quite on the contrary: Non-TEP's are easy to spot and exclude, yet he probably overlooks some real TEP's due to the variant grammatical structures or tiredness. Then we apply our method in Chen's solution to annotating Chinese patent data[3]. With a little human involvement in excluding obvious non-TEP's, our method is able to complete in a few seconds what takes a human several days.

When the algorithm runs without DDC, the recalls are extremely low. Once a 3% abstracts are included in DDC, an obvious increase is shown in all recalls. As more abstracts are included, the increment rates drop, and the growth basically stops when recalls reach over 80%. This indicates a lower bound for the amount of manual extraction. On the other hand, precisions increase only in the beginning[2]. Then they are on a steady decrease because manual extraction introduces "noises" to DDC. Thus an upper bound of manual extraction amount is expected. This relationship is better seen from the F-Measures. In each column there is a peak for F-Measure. The corresponding M.E. value is what we recommend as the ideal manual extraction amount in each domain. Interestingly, the more patents in a domain, the less the percentage of manual extraction is recommended, and our method is therefore more cost-effective.

[2] This increase is caused by the inclusion of more correct TEP's in the result set, not the exclusion of the wrong ones.

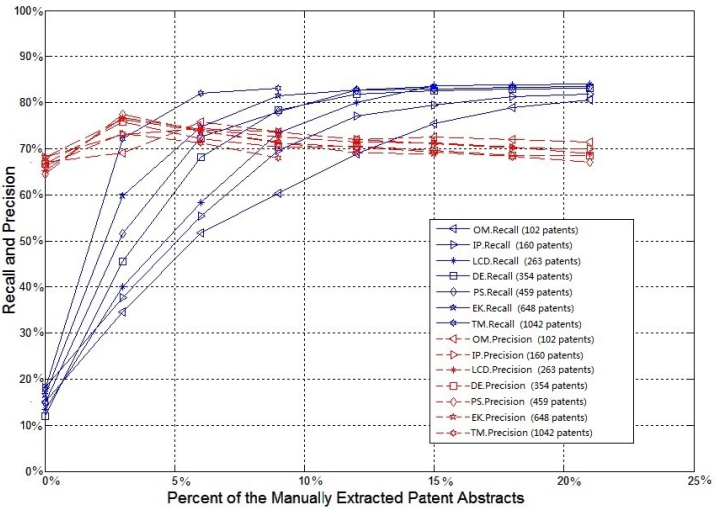

Fig. 6. Variance of precision and recall. The numbers in the brackets on the legend are the number of patents in the domains.

4.3 Determining the Number of Abstracts to Manually Extract

Formula (1) in Section 3.4 helps a patent analyzer to decide how many patent abstracts to manually extract in a certain domain. Now we discuss in detail how the function is determined.

In Table 2 we see that more patent abstracts in a domain means less *portion* to manually extract, but the *number* increases while the increment rate drops. Thus a logarithm function $y = aln(x) + b$ is a good approximation, where y is the number of manually extracted abstracts and x is the total patent number. Its parameters are determined as follow: Denote $ln(x)$ by t. For each domain in Table 2, take the natural logarithms of the total abstract number as a value of t, take the product of the "ideal portion" and total patent number as a value of y. Then we use Least Square Fit to determine a and b for function $y = at + b$. The result $y = 19ln(x) - 66$ is plotted in Figure 7.

4.4 Comparing with Other Methods

We compare our method with two standard keyword extraction algorithms: the TF-IDF algorithm and Matsuo's algorithm[6] for extracting keywords in a single document. We merge all abstracts of a domain into a single file as the input for Matsuo's statistical method, and segment all sentences into words by ICTCLAS[10]. Finally we choose the top 10 words extracted from each abstract by TF-IDF and top 4000 words extracted from the merged file by Matsuo's method. The result is shown in Figure 8. Our method outperforms the two algorithms in both precision and recall. The poor performance of the first two algorithms is mainly a result of their inherent ineligibility. Though they have

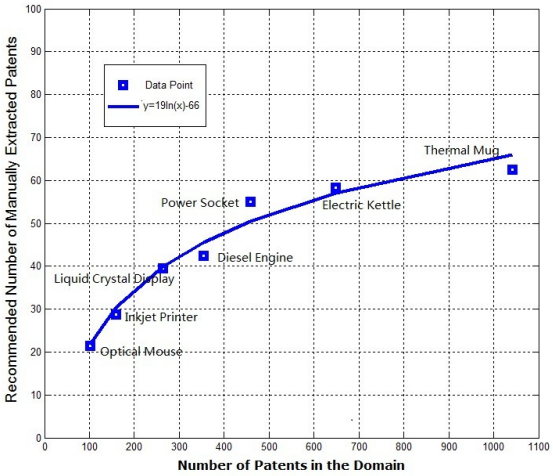

Fig. 7. Graph of $y = 19ln(x) - 66$. Domain names are given beside each data point.

been modified to adapt to TEP extraction, the targeted keywords are not necessarily TEP's. The two algorithms are also impaired by the inaccuracy in word-segmentation by ICTCLAS. In contrast, our method is free from such influences of word tokenizer.

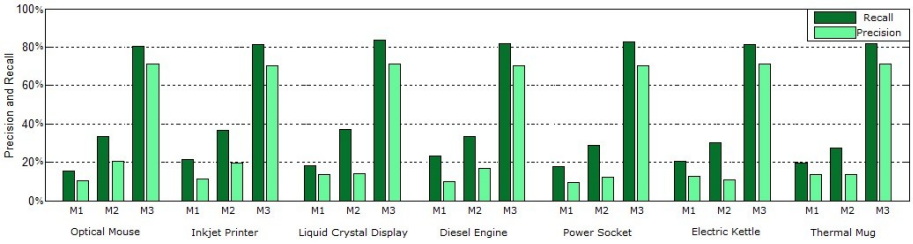

Fig. 8. Comparison in recall and precision. M1 is TF-IDF, M2 is Matsuo's algorithm[6], and M3 is our method.

5 Conclusions

Technology effect phrase (TEP) extraction is an indispensable part of many patent mining algorithms. To reduce human labor in the extraction, we propose a method based on partitioning corpus to extract TEP's from Chinese patent abstracts. To further increase precision, we propose 3 rules to deal with some specific yet common cases. We also find that the larger the patent domain, the more cost-effective the method. We give a formula to calculate the optimal number of patent abstracts to manually extract. The logarithm function indicates that human workload will be acceptable even when the number of patents is enormous.

In future, we will try to further increase the recall of our method. According to our experiments, adding more patent abstracts into DDC alone does not help, because the

F-Measure will drop fast due to the increasing loss of precision. How to increase recall without sacrificing precision will be our future work.

Acknowledgement. This work is supported by the Key Program of National Natural Science Foundation of China (61232002) and the National "863" High-tech Research Development Plan Foundation (2012AA011004).

References

1. Alberts, D., et al.: Introduction to patent searching. In: Currentchallenges in Patent Information Retrieval, pp. 3–43. Springer, Heidelberg (2011)
2. Chen, Y., Zhang, X.: Research on the identification of technology effect phrases in patents. Contemporary Technology in Library and Information Science 12, 010 (2011)
3. Chen, X., Peng, Z., Zeng, C.: A co-training based method for chinese patentsemantic annotation. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management. ACM (2012)
4. Nanba, H., Fujii, A., Iwayama, M., Hashimoto, T.: Overview of the patent mining task at the NTCIR-8 workshop. In: Proceeding of the Eighth NTCIR Workshop, Tokyo, Japan, June 15-18, pp. 293–302 (2010)
5. Parapatics, P., Dittenbach, M.: Patent claim decomposition for improvedinformation extraction. In: Current Challenges in Patent Information Retrieval, pp. 197–216. Springer, Heidelberg (2011)
6. Matsuo, Y., Ishizuka, M.: Keyword extraction from a single document using wordco-occurrence statistical information. International Journal on Artificial Intelligence Tools 13(01), 157–169 (2004)
7. Ni, W., Liu, T., Zeng, Q.: An Under-Sampling Approach to ImbalancedAutomatic Keyphrase Extraction. In: Gao, H., Lim, L., Wang, W., Li, C., Chen, L. (eds.) WAIM 2012. LNCS, vol. 7418, pp. 387–398. Springer, Heidelberg (2012)
8. Levy, R., Manning, C.: Is it harder to parse Chinese, or the Chinese Treebank? In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, vol. 1. Association for Computational Linguistics (2003)
9. The Stanford Parser: A statistical parser, `http://nlp.stanford.edu/software/lex-parser.shtml`
10. Zhang, H.-P., et al.: HHMM-based Chinese lexical analyzer ICTCLAS. In: Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, vol. 17. Association for Computational Linguistics (2003)
11. Yuuji, K.: Inkjet Printer: China, 02154598.7[P]. 2003-06-18
12. Xue, X., Bruce Croft, W.: Automatic query generation for patent search. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management. ACM (2009)