Transdisciplinary Engineering for Complex Socio-technical Systems
K. Hiekata et al. (Eds.)

doi:10.3233/ATDE190158

Using Machine Learning Approach to Identify Synonyms for Document Mining

Amy J.C. TRAPPEY[a,1], Charles V. TRAPPEY[b], Jheng-Long WU[c] and Kevin T.-C TSAI[a]

[a] *Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Taiwan*
[b] *Department of Management Science, National Chiao Tung University, Taiwan*
[c] *School of Big Data Management, Soochow University, Taiwan*

Abstract. Technical or knowledge documents, such as research papers, patents, and technical documents, e.g., request for quotations (RFQ), are important knowledge references for multiple purposes. For example, enterprises and R&D institutions often need to conduct literature and patent searches and analyses before, during, and after R&D and commercialization. These knowledge discovery processes help them identify prior arts related to the current R&D efforts to avoid duplicating research efforts or infringing upon existing intellectual property rights (IPRs). It is common to have many synonyms (i.e., words and phrases with near-identical meanings) appeared in documents, which may hinder search results, if queries do not consider these synonyms. For instance, conducting "freedom-to-operate" (FTO) patent search may not find all related patents if synonyms were not taking into consideration. This research develops methodologies of generating domain specific "word" and "phrase" synonym dictionaries using machine learning. The generation and validation of both domain-specific "word" and "phrase" synonym dictionaries are conducted using more than 2000 solar power related patents as testing document set. The testing result shows that, in the solar power domain, both word level and phrase level dictionaries identify synonyms effectively and, thus, significantly improve the patent search results.

Keywords. Synonym Extraction, Machine Learning, Pattern-based Extraction, Self-supervised Learning.

Introduction

Companies need to conduct patent searches regularly before their products being launched (i.e., FTO) or before carrying out any R&D activities. The search helps identify which prior art IPs overlap with specific technologies being developed to prevent infringing upon existing IPRs [1]. However, current patent laws do not explicitly restrict standard terminologies to be used for particular domains when writing patent applications. Rather, IP offices allow patent applicants to write their applications freely using synonyms, which may cause incomplete or inaccurate patents found using traditional search queries [2].

Traditional patent searches use the exactly matching "key terms" Boolean query algorithm, which often miss out documents consisting of synonymous words. This study

[1] Corresponding Author, Email: trappey@ie.nthu.edu.tw.

aims to construct a smart patent search that considers synonyms. First, the proposed word and phrase synonym dictionaries are constructed based on a traditional OPTED dictionary and a web encyclopedia corpus. Although the terminologies have many synonymous terms, they can be found referring to the newly built word and phrase synonym dictionaries. The synonym-enhanced search approach will be smarter than the original search/query method and helps companies conduct comprehensive patent search and analysis to avoid possible threats of IP disputes. This research will also conduct a search test in a specific domain of solar power related patents and compare the differences between the intelligent search method proposed in this research and the traditional method.

1. Literature review

The relevant literature are divided into four parts. The word embedding and vectorization explain the different methods and concepts of projecting words into vector space. The second part is synonym extraction, which describes various methods and techniques for extracting synonyms in the text. The third part will discuss the properties of the corpus used in this study. The final section is the ontology schema describing methods and algorithms for conducting synonym extraction.

1.1 Word embedding

In Natural Language Processing (NLP), word embedding is a widely used technique. Word embedding means that words are mapped into space, so the sentences can be expressed by vectors. Those vectors can be imported into the model for the mathematical calculations.

One-Hot Encoding is the simplest word vectorizing method. Each element in the vector represents the presence or absence of a word in the corpus vocabulary. If a sentence contains a word, the element corresponding to the word is 1; otherwise, the element corresponding to the word is 0. This method cannot reflect the similarity between two words. The order of the words in a sentence are also not considered. All the words in the document are independent and are not related to other words. For example, two sentences "Mary likes to sing" and "Mike likes to dance" can be used to create a dictionary {"Mary": 1, "likes ": 2, “to”: 3, “sing”: 4, “Mike”: 5, “dance”: 6}. In One-Hot Encoding, the two sentences are expressed in the form of two vectors, [1, 1, 1, 1, 0, 0] and [0, 1, 1, 0, 1, 1], where all sentences or documents are of equal length. Since the lengths of word vectors are all equal to the vocabulary size, a major drawback of this method is that the data is sparse and easily leads to curse of dimensionality [3].

Compared with One-Hot Encoding, Term Frequency-Inverse Document Frequency (TF-IDF) considers more information of sentences or documents. In TF-IDF, TF stands for term frequency, that is the frequency that a particular word appears in a particular document. The more frequently the word appears in a document, the higher TF value of the word is. IDF indicates the inverse document frequency. IDF is to represent the unique or distinguishing feature of a word. A word appears in fewer documents, the higher its IDF value is [4]. If a word appears multiple times in a document and rarely appears in other documents, the word will have a high TF-IDF weight and is more representative of the document. The TF-IDF equation is derived as follows.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (1)$$

$$idf_i = log\frac{|D|}{|\{j:t_i \in d_j\}|} \quad (2)$$

$$tfidf_{i,j} = tf_{i,j} \times idf_i \quad (3)$$

Where $tf_{i,j}$ represents term frequency. $n_{i,j}$ shows the number of occurences of term i in document d_j. $\sum_k n_{k,j}$ is the total number of occurrences of all words in document d_j, $|D|$ is the number of files in the corpus, $|\{j: t_i \in d_j\}|$ is the number of documents which contain t_i.

Word2vec was proposed in 2013. Considering the order of the context, the words are projected into the vector space. There are two types of model in Word2vec. The first model called Continuous Bag-of-Words Model (CBOW), which predicts the center word by the words around it. That is, predicting W_t by W_{t-2}, W_{t-1}, W_{t+1}, W_{t+2} and so on. The other is Continuous Skip-gram Model. In contrast to CBOW, it uses the center word to predict the surrounding words. That is, predicting W_{t-2}, W_{t-1}, W_{t+1}, W_{t+2} by W_t [5]. CBOW and Skip-gram have better performance than other Neural Network Language Models (NNLMs) in semantic accuracy or syntactic accuracy. Word2vec is also an important foundation for the future development of Doc2vec [6].

1.2 Naive Bayes classifier

Naive Bayes classifier is a probabilistic classification method based on Bayes' theorem, assuming that features are independent. But in the study of Domingos and Pazzani [7], the experiment shows that although the features are strongly correlated, the classifier can still have good classification performances. The simple Bayesian classifier is often used in applications such as automatic document classification.

Bayes' theorem proposed by the 18th-century British mathematician, Thomas Bayes. The main idea of Bayes' theorem is to modify the old ideas with the latest evidence. If B event occurred, the probability of the occurrence of event A is as follows

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (4)$$

Similarly, under the condition that the A event occurs, the probability of the occurrence of event B is

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \quad (5)$$

By (4) and (5)

$$P(A|B)P(B) = P(B|A)\,P(A) = P(A \cap B) \quad (6)$$

Both side of equation (6) divided by P(B). If P(B) is non-zero, then Bayes' theorem is obtained as follows

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \quad (7)$$

In equation (7), P(A) is also called the prior probability, which represents the probability distribution of A before the observation of evidence B and does not consider any factors related to B; P(A | B) called posterior probability, which means the conditional probability obtained after observing evidence B.

Naive Bayes classification applies Bayes' theorem to perform classification. In the application of document classification, given a set of vectorized keyword frequencies, the probability of a certain keywords belonging to a certain type of documents can be calculated by Bayes' theorem

$$P\left(class_i \middle| \overrightarrow{tf}\right) = \frac{P\left(\overrightarrow{tf} \middle| class_i\right) P(class_i)}{P(\overrightarrow{tf})} \tag{8}$$

$class_i$ means i-th category, $\overrightarrow{tf}$ is the vector of the keyword frequency, Since $P\left(\overrightarrow{tf}\right)$ in each class is always same, the classifier trains $P\left(\overrightarrow{tf} \middle| class_i\right)$ to make classification. In the application of document classification, the simple Bayesian classifier often uses Bernoulli Naive Bayes and Multinomial Naive Bayes. The former only considers whether the keywords appear or not and denoted as 1 or 0. The latter consider the term frequency. For example, assume that we want to classify some articles into sport news or financial news. The article that contains many words related to basketball will be classified as sport news rather than financial news because $P\left(\overrightarrow{tf} \middle| sport\ news\right)$ is much larger than $P\left(\overrightarrow{tf} \middle| financial\ news\right)$. Bayesian classification has good classification ability under few samples. Even the training set is insufficient, the classification accuracy is still good [8].

1.3 Synonym extraction

In synonymous word extraction, Inverted Index Extraction (IIE), Pattern-based Extraction (PbE) and MaxEnt are three methods that able to capture synonymous words by the characteristics of the dictionary. The former two methods are rule-based algorithms and the latter is a machine learning method. The final experimental results show that PbE has the best results. Each dictionary has different narrative habits and formats. PbE sets an initial feature to extract synonyms, and then search the synonyms reversely. The features are iteratively obtained by bootstrapping. It is possible to find new features in every iteration. Finally, extracted synonyms must be filter by the transitive closure. If the synonyms form a loop with each other, they will be reserved; otherwise, if a synonym cannot be linked back to the original word, it will be deleted [9].

In synonymous phrase extraction, a self-supervised learning method has been proposed. This method automatically generates labeled training samples and segment the sentences with Conditional Random Field (CRF). CRF is a discriminative probability model. The parameters considered are sentence s, word position i, current word label type I_i, and previous word label type I_{i-1}, as in equation (1). By establishing several characteristic functions such as equation (2), the weight of each feature is obtained through training [10].

$$\text{score}(\text{l}|\text{s}) = \sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(s, i, l_i, l_{i-1}) \tag{9}$$

$$f_j(s, i, l_i, l_{i-1}) \quad , j = 1,2, \ldots, m \tag{10}$$

The noun entities within a certain distance will be selected as synonym candidates [11]. Wikipedia contains some useful structural data, including internal links, redirects, dis-ambiguations, and categories [12][13][14][15]. We extract synonymous phrases from Wikipedia contain. Positive samples are collected by many redirect pages (redirection pages) and Named Entity Recognition (NER) is used to collect the negative samples.

1.4 Patent search

Nowadays, there are many patent search platforms. Some of them are free such as Google Patents, United States Patent and Trademark Office (USPTO), Free Patents Online (FPO), etc. There are also some paid platforms such as WIPS, PatBase, Derwent Innovation (DI), etc. USPTO provides quick search and advanced search. Users can set the conditions to find the patents. USPTO only provides exactly matching and the number of results is limited. FPO has a basic word stemming function and Google Patents show the results with some basic statistics such as top assignees and inventors. The major defects of these free patent search platforms are the searching flexibility and it's hard to export and analyze the results.

WIPS is the first online worldwide patent information service provider in South Korea. PatBase was created by Minesoft and RWS. In WIPS and PatBase, users can combine the results their got in deferent queries to get new results and there are more specific conditions to choose while searching. Derwent Innovation (DI) database consolidates patents from more than 90 country/regional patent offices. In DI, smart search finds keywords from a paragraph of text, provided by the users, then the patents are identified containing these keywords. Furthermore, the wildcard and proximity operators improve the searching results in DI and WIPS. Although the paid patent platforms seem smarter than the free platforms, the synonyms they can identified to enhance search versatility is still limited. This research intends to significantly enhance the comprehensive patent search results.

1.5 Ontology map

Figure 1 shows the ontology schema of hierarchically presented key parts for synonym extraction. Synonym extraction methods (in both word and phrase levels) and the word and phrase corpuses as synonym training bases are the critical parts in the ontology. There are some sub-techniques for word level synonym extraction, e.g., Inverted Index Extraction (IIE), Pattern-based Extraction (PbE), and Maximum Entropy (MaxEnt). For phrase level synonym extraction, algorithms such as Entity Link Frequency (ELF), Average Cosine Similarity (ACS), Pseudo Relevance Feedback (PRF), Average Ranking (AveR), Ranking Score Combination (RSC), and Self-supervised synonym extraction are highlighted [16]. The main training corpuses are The Online Plain Text English Dictionary (OPTED), Dictionary.com, and Wikipedia [17].

2. Methodology

This study proposed a method of detecting synonyms of key terms (both words and phrases) for smart patent search. The research process can be divided into several parts. First, a synonym dictionary must be constructed for synonymous words. This step is completed by feature extraction. By continuously iterating by observing the grammatical features, more features can be found to capture the synonym. We use OPTED as our corpus.

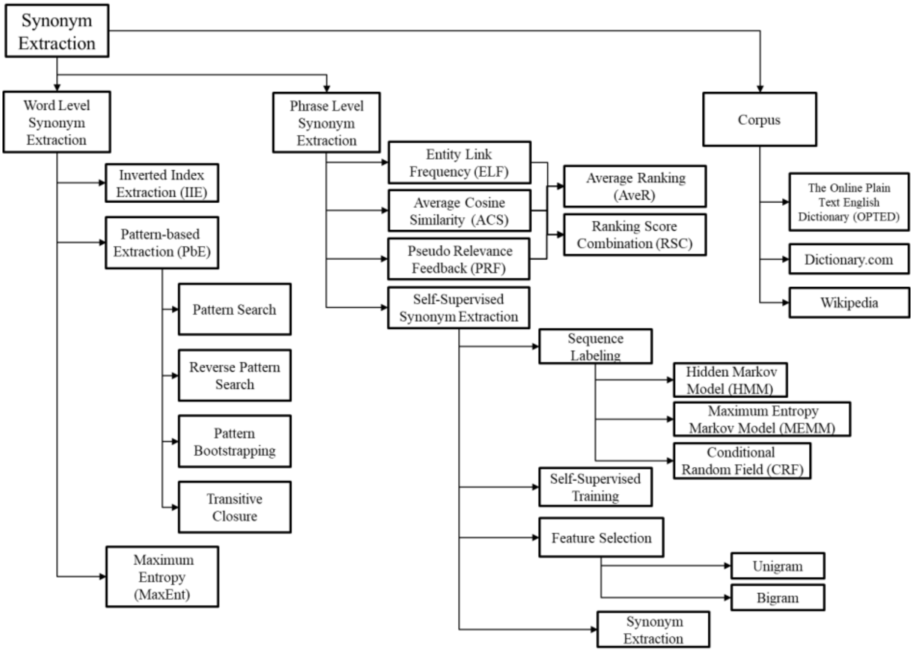
Synonym Extraction
Word Level Synonym Extraction
Phrase Level Synonym Extraction
Corpus
Inverted Index Extraction (IIE)
Pattern-based Extraction (PbE)
Pattern Search
Reverse Pattern Search
Pattern Bootstrapping
Transitive Closure
Maximum Entropy (MaxEnt)
Entity Link Frequency (ELF)
Average Cosine Similarity (ACS)
Pseudo Relevance Feedback (PRF)
Self-Supervised Synonym Extraction
Average Ranking (AveR)
Ranking Score Combination (RSC)
Sequence Labeling
Hidden Markov Model (HMM)
Maximum Entropy Markov Model (MEMM)
Conditional Random Field (CRF)
Self-Supervised Training
Feature Selection
Unigram
Bigram
Synonym Extraction
The Online Plain Text English Dictionary (OPTED)
Dictionary.com
Wikipedia

Figure 1. The ontology map outlines the key parts of synonym extraction methods and corpuses.

The second part is to construct a synonym dictionary for finding synonymous phrases. The synonym of this part contains more professional nouns or multiple words. For example, World Trade Organization and WTO are synonymous phrases. The first step is sequence labeling. We train the model with Conditional Random Field (CRF). After the sequence labeling process, in all articles, each two noun phrases within a certain distance are a candidate for synonymous phrases. We use Multinomial Naïve Bayes Classifier to classify two nouns as synonyms or not. Our positive samples are collected by many redirect pages (redirection pages) in Wikipedia. Negative samples are more complex, by determining several heuristic rules, such as subsets and supersets, categories, etc.

After the first two parts are completed, the two dictionaries are merged into one. If a user input a query string, after the program accepts the string, it compares it with the current synonym dictionary. The program then adds all synonymous words or phrases to the search string with an "OR" condition. Then we use Derwent Innovation (DI) to get thousands of patents in a certain domain to validate the efficiency of the program. In the validation phase, we test the precision by TOEFL Synonym Questions2 and test the recall by comparing with a dictionary called Synonyms and Antonyms: An Alphabetical List of Words in Common Use (ALWC). To construct a word level synonym dictionary, we use an algorithm similar to Pattern-based extraction (PbE) and the source of the data is OPTED. We use the regular expression to define the patterns.

After the data preprocessing, the first step is to establish the initial feature. Most of the dictionaries have a "description part" and a "synonym part" and they will have its own unique syntax. The most common feature of a synonym is behind a semicolon. For

[2] Synonym Dataset: TOEFL Synonym Questions, http://lsa.colorado.edu/

example, the interpretation of bag "a container or receptacle of leather, plastic, cloth, paper, etc., capable of being closed at the mouth; Pouch.” It can be seen that pouch is a synonym for bag, so this feature can be used as an initial feature to extract more synonyms. Then the reverse searching step starts. For example, the interpretation of stain is “a cause of reproach; stigma;” so stigma is the synonym of stain. the interpretation of tarnish is “to diminish or destroy the purity of; stain; sully.” Now, stain appears as a synonym of tarnish so tarnish will also be added in to the synonyms of stain.

Bootstrapping will be implemented after the previous step. The algorithm will search for new features in the dictionary based on old patterns and the features will continue to increase iteratively. For example, we assume that tarnish is our target word. In its interpretation “to diminish or destroy the purity of; stain; sully” sully will be added in to the synonym of tarnish. In the interpretation of sully “to soil, stain, or tarnish.” Tarnish appears in another pattern “to A, B or C” so this will be the new pattern. Finally, in the transitive closure phase, in order to ensure that the synonyms are indeed highly similar, the algorithm will test whether synonyms can form a transitive closure. If the synonyms cannot form a loop, it will be dropped.

The word level synonym dictionary (original version) generated by this method is not as good as expected. Therefore, this study improves it from three directions. The first one is the expansion of part of speech; The second direction is to enhance the inference ability. Finally, the third approach is to add more undiscovered important features. The feature number increased from 6 to 12. The methods can effectively improve the efficiency of synonym extraction. Table 1 shows the patterns of word level dictionary. Pattern 7 to 12 are the differences between two versions.

Table 1. Example patterns of word level dictionary.

Pattern no.	Pattern	Example sentence	Synonym found
1	"^.*; (\w+).$"	Abrogate (a.) Abrogated; abolished.	abolished
2	"^.*; (\w+);"	Abbreviate (a.) Abbreviated; abridged; shortened.	abridged
3	"^.*; the (\w+).$"	Acme (n.) The top or highest point; the culmination.	culmination
4	"^.*; to (\w+).$"	Abalienate (v. t.) To transfer the title of from one to another; to alienate.	alienate
5	"^.*; a (\w+).$"	Abider (n.) One who dwells; a resident.	resident
Notes:	^ | $ | . | *	Beginning of line | end of line | any single character match | the preceding element match zero or more times	

This study refers to and modifies the self-supervised learning algorithm mentioned above and improves the details of collecting negative training samples and the classifier to construct a phrase level synonym dictionary. There are four main steps: data preprocessing, collecting positive training samples, collecting negative training samples, and training and classification. n this study, the corpus used to construct the phrase level synonym dictionary is Wikipedia. The data we used is the enwiki dump provided by Wikimedia Foundation (WMF). Since the file is a structured XML file with many tags, such as <page>, <revision>, <title>, <id>, <redirect>, <text>, <ref>, etc. We use Python to remove labels, symbols, and noise to produce csv files for all Wikipedia articles. In addition to them, this XML file also contains a lot of structured information like the paired list of redirect pages.

Before training the classifier, we must have enough positive and negative training samples. A positive training sample refers to the sentence between the two synonyms and its label is set to 0. A negative training sample is the sentences between two non-synonyms and labeled as 1. In Wikipedia, redirect pages is a type of special pages that users often redirected to a redirect page when searching for many similar keywords. These keywords and redirected pages usually express the same or very similar entities. We use this Wikipedia's structural features to collect positive training samples. In the previous phase, all pairs of redirect pages have been extracted. In this step, for each "redirected to" and "redirected from" page, if they appear in a sentence within a distance. The text between them will be regarded as a positive training sample. Table 2 shows some of the positive training samples. The first column are the pages redirected from, and the second column are the pages redirected to. The third column are the sentence between them, not including the page name. For example, in the first row, the original sentence is “Anarchist movement as known by the Anarchism…” and the page is redirected from Anarchist movement to Anarchism in Wikipedia.

Table 2. Sample texts between positive terms using Wikipedia’s “redirect pages” (e.g., term 1 and term 2 synonym pairs) as training base.

Term 1 (redirected from)	**Term 2 (redirected to)**	**Identify words/phrases/symbols between two terms**
Anarchist movement	Anarchism	as known by the
Oscars	Academy Awards	also known as the
ANSI	American National Standards Institute	(
Induced abortion	Abortion	is often used to mean only
Australian Football	Australian rules football	officially known as

Negative samples are collected by Named Entity Recognition (NER) in spaCy. If two nouns appear within a certain distance at the same time, and their named entity are different. The sentences between them will be added in to negative training samples. Table 3 shows part of negative training samples. For example, in the first row, the original sentence is “…the NL Division Series, and then the Atlanta Braves…”. The first noun, the NL Division Series and the second noun, the Atlanta Braves are different types of noun and they appear within a certain distance so the text between them are collected as a negative training sample, which means they are not synonyms.

Table 3. Samples of text between negative pairs trained based on NER characteristics of adjacent terms.

Term 1	**Term 2**	**The text between two terms**
the NL Division Series	the Atlanta Braves	, and then
Randy Johnson	Curt Schilling	and
23-Mar-07	New York City	, in
French	Quebec	or
New York City	1931	and founded in

We use 2000 positive samples and 2000 negative samples for the training data. First, we use the CountVectorizer in scikit-learn to vectorize the training data and send it into the Multinomial Naive Bayes classifier for training. The Multinomial Naive Bayes classifier is suitable for the classification of discrete features, such as the word frequency. This study uses 4,000 training samples. After the training phase, the classifier can classify unknown sentences. In all articles in Wikipedia, we collected all pairs of nouns within a certain distance and then put them into the pre-trained classifier for prediction so they can be classified into synonyms or non-synonyms. Finally, all predicted synonyms are collected to form a phrase level synonym dictionary. Table 4 shows samples of word/ phrase level synonym pairs automatically extracted in this research.

Table 4. Samples of word and phrase synonym pairs.

Word/phrase 1	**Word/phrase 2**
consequence	outcome
generate	produce
blaze	flame
JAIR	Journal of Artificial Intelligence Research
ANOVA	Analysis of variance
TV	Television

3. Validation

First, we test the synonym word dictionary using TOEFL synonym questions to measure the proposed method's accuracy in identifying synonyms. There are 80 questions and each question is a multiple choice question (choose one from four). Only one option is synonymous or very similar to the word of the question. The final version of our synonym dictionary gets 56 right answers, reaching 70% accuracy. The validation results show that the final version of the word level synonym dictionary can identify a high proportion of synonyms. Recall is the proportion of real positive cases that are correctly predicted positive. This research further tests the recall capability of the proposed methods by comparing our method with an existing dictionary, ALWC. We collect 2,280 patens in solar power domain to be the testing corpus. For example, 365 and 124 synonyms in word and phrase levels respectively are found using our dictionaries, while only 88 synonyms are found using ALWC dictionary in patent #1. On average, 190 and 67 word and phrase synonyms found using our dictionaries for each patent. On average, only 57 synonyms are found using ALWC dictionary in each patent. Over all, our word and phrase level dictionaries outperform the existing ALWC dictionary. Table 5 shows some samples of synonyms found in a sample patent (#167).

Table 5. Samples of synonyms found in the abstract of patent no. 167.

Phrase level synonym pairs	**Word level synonym pairs**
AC-to-DC conversion, rectifier	rectificator, rectifier
petty patents, utility model	subvert, evert
marine water, seawater	wreathe, twine
electrical impulses, electricity	severally, individually

4. Conclusion

In this paper, to conduct a complete freedom-to-operate analysis before the product put on the market, we constructed both word level dictionary and phrase level dictionary by pattern-based method and machine learning approach. The proposed method has made significant performance for synonym mining of technical document. The test result shows that the phrase level dictionary is able to detect the technical terms in many synonymous forms and the word level dictionary outperforms the existing dictionary. The patent search can be more complete through the dictionaries we constructed, and the freedom-to-operate analysis can be improved. The potential intellectual property litigations and disputes can be avoided.

Acknowledgement

This research is partially supported by the Ministry of Science and Technology research funding (MOST-107-2221-E-007-071 and MOST-107-2410-H-009-023) in Taiwan.

References

[1] Avoiding patent infringement using freedom-to-operate analyses, retrieved from https://www.tilleke.com/resources/avoiding-patent-infringement-using-freedom-operate-analyses, 2016.

[2] B. Yoon and P. Yongtae, A text-mining-based patent network: Analytical tool for high-technology trend, *The Journal of High Technology Management Research*, 2004, 15(1), pp. 37-50.

[3] J. Turian, L. Ratinov and Y. Bengio, Word representations: a simple and general method for semi-supervised learning. In Proceedings of the 48th annual meeting of the association for computational linguistics. *Association for Computational Linguistics*, 2010, pp. 384-394.

[4] G. Salton, E. A. Fox, and W. Harry, Extended Boolean information retrieval, *Communications of the ACM*, 1983, Vol. 26, Issue 11, pp. 1022-1036.

[5] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, Distributed representations of words and phrases and their compositionality, *Advances in neural information processing systems*, 2013, pp. 3111-3119.

[6] Q. Le, and T. Mikolov, Distributed representations of sentences and documents, *International conference on machine learning*, 2014, pp. 1188-1196.

[7] P. Domingos, and M. Pazzani, Beyond independence: Conditions for the optimality of the simple bayesian classifier. *Proc. 13th Intl. Conf. Machine Learning*, 1996, pp. 105-112

[8] A. McCallum, and K. Nigam, A comparison of event models for naive Bayes text classification, *AAAI-98 workshop on learning for text categorization*, 1998, pp. 41-48.

[9] T. Wang, and H. Graeme, Exploring patterns in dictionary definitions for synonym extraction, *Natural Language Engineering*, 2012, 18(3), pp. 313-342.

[10] E. B. Ordway, *Synonyms and antonyms: an alphabetical list of words in common use grouped with others of similar and opposite meaning*, George G. Harrap, London, 1918

[11] H. Wu, and Z. Ming, Optimizing synonym extraction using monolingual and bilingual resources, Proceedings of the second international workshop on Paraphrasing-*Volume 16, Association for Computational Linguistics*, 2003, pp. 72-79.

[12] A. Jagannatha, C. Jinying, and Y. Hong, Mining and ranking biomedical synonym candidates from Wikipedia, *Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis*, 2015, pp. 142-151.

[13] A. Krizhanovsky, Synonym search in wikipedia: Synarcher, *arXiv* preprint cs/0606097, 2006.

[14] C. Lee, S. Bomi, and P. Yongtae, How to assess patent infringement risks: a semantic patent claim analysis using dependency relationships, *Technology Analysis & Strategic Management*, 25(1), 2013, pp. 23-38.

[15] R. Bunescu, and P. Marius, Using encyclopedic knowledge for named entity disambiguation, *11th conference of the European Chapter of the Association for Computational Linguistics*, 2006, pp. 9-16.

[16] F. Hu, S. Zhiqing, and R. Tong, Self-Supervised Synonym Extraction from the Web, *J. Inf. Sci. Eng.*, 31(3), 2015, pp. 1133-1148.

[17] C. Bøhn, and N. Kjetil, Extracting named entities and synonyms from Wikipedia, *Advanced Information Networking and Applications (AINA)*, 2010, pp. 1300-1307.