Exploring Evolutionary Technical Trends From Academic Research Papers

Teng-Kai Fan, Chia-Hui Chang
Department of Computer Science and Information Engineering,
National Central University, Chung-Li, Taiwan 320, ROC
tengkaifan@gmail.com, *chia@csie.ncu.edu.tw*

Abstract

Automatic Term Recognition (ATR) is concerned with discovering terminology in large volumes of text corpora. Technical terms are vital elements for understanding the techniques used in academic research papers, and in this paper, we use focused technical terms to explore technical trends in the research literature. The major purpose of this work is to understand the relationship between techniques and research topics to better explore technical trends. We define this new text mining issue and apply machine learning algorithms for solving this problem by (1) recognizing focused technical terms from research papers; (2) classifying these terms into predefined technology categories; (3) analyzing the evolution of technical trends. The dataset consists of 656 papers collected from well-known conferences on ACM. The experimental results indicate that our proposed methods can effectively explore interesting evolutionary technical trends in various research topics.

1. Introduction

The importance of document understanding has increased due to the large volume of texts published that researchers must search through to find relevant information. Various approaches have been proposed to assist in this task: text summarization, information extraction, and others. For example, text summarization is a main task in DUC that aims to represent a document in a short paragraph or few sentences. Information extraction has also been applied in financial news to find management succession events. In a sense, information extraction is a special type of text summarization that is designed specifically for the input documents. Thus, in this paper we consider what is the proper task to be used for academic research papers?

For many novice graduate students who begin their study in a research field, understanding common techniques in that field is crucial. For example, *TF-IDF* (term frequency and inverse document frequency) is a commonly used weighting technique for information retrieval, while decision trees and Naïve Bayes are two common algorithms for classification. For experienced researchers with some background knowledge in a research field, comparing different techniques used for a particular research problem might help to exhaustively explore approaches for such problems. As time axis extends, the relationship between these techniques and research topics may also provide insight to the evolutionary technical trends. In any case, it would be very to automatically explore, recognize, and summarize trends in research techniques.

Despite its importance, however, automatically development trends in academic research papers have not been well addressed in the existing works. Some existing text mining methods only discover the theme patterns but do not investigate the technical trends [5][6][10][12]. In this study, we define three tasks for exploring evolutionary technical trends in academic research papers: (1) recognizing focused technical terms from research papers; (2) classifying these terms into predefined categories of technology; and (3) analyzing evolutionary technical trends. For this, we define focused technical terms for a research paper as terms for techniques "**applied"** in a research paper (i.e., excluding terminologies that are mentioned but not actually employed).

We evaluated the proposed methods on ACM (Association for Computing Machinery) papers. The data consisted of 656 papers on two research topics, SIGIR (Special Interest Group on Information Retrieval) and KDD (Knowledge Discovery and Data Mining), from 2002 through 2006. Every focused technical term is assigned to one of ACM's classification names. The experimental results indicate that our proposed methods can effectively explore noteworthy technical trends in various research topics by visual representations.

The remainder of this paper is organized as follows: Section 2 and Section 3 describe our proposed

approach for focused technical term recognition and term classification respectively. The experiments are presented in Section 4. Section 5 introduces related works that, including an overview of term variation, terminology searching, term classification and topic pattern detection. Finally, we present conclusions in Section 6.

2. Term Recognition

Before demonstrating the proposed methods, we will explain what focused technical terms are since the task of focused technical term extraction is slightly different from general term recognition. Any given scientific study may involve with several technical terms, such as the approach proposed by the authors, well-known datasets, existing techniques and evaluation measures. Nevertheless, we only define focused technical terms as those techniques applied in a research paper. Thus, even if several terminologies are mentioned in a study, we are only concerned with those that are really employed. For example, in the sentence, "The main reason that we chose *XML* rather than *MathML* is due to its extensibility", two technical terms, *XML* and *MathML*, are mentioned; however, only the term *XML* is considered a focused technical term (henceforth called term), while *MathML* is only noted for comparison.

Given a paper as input, we conduct three steps before the learning step: text pre-processing, candidate term selection, and feature construction.

2.1 Text Pre-Processing

We observe that the non-focused technical terms sometimes occur in a sentence containing negative words (or comparative words), such as *different from*, *instead of*. To detect a sentence including negative words, we consult the sentiment dictionary [19] that collects different types of sensitive words to define the initial negative word sets manually. Since a small dictionary may have the problem of coverage, a popular thesaurus (e.g., WordNet [20]) is used to enlarge the initial negative word sets. We submit each negative/comparative word included in the sentiment dictionary to WordNet to obtain corresponding synonyms. For instance, the comparative word *different* has the synonym *dissimilar*. Hence, each sentence that contains one or more negative words (or comparative words), will be removed to avoid the extraction of non-focused technical terms.

A concept can be linguistically illustrated by any of the surface representations. In order to maintain implications of a term, we consider orthographic (e.g., the usage of hyphens, slashes and asterisk) and morphological (e.g., singular and plural) aspects to define our canonical form. In other words, we normalize each word in a singular form and remove the special characters, e.g.:

orthographic: tf-idf → tf idf.
morphological: search engines → search engine.

Additionally, since academic research papers usually contain meaningless content in fixed positions, such as page numbers, the Affiliation section, the References section and the Acknowledgements section, we perform a heuristic mechanism to remove these irrelevant contexts.

2.2 Candidate term selection

According to [3][4], three common selection approaches have been used for candidate term selection: *n*-grams, NP-chunks, and Pos Tag Patterns. Nakagawa [6] points out that in technical documents most domain-specific terms are noun phrases or compound nouns. We also note that the potential terms are partly four tokens or more in length. Hence, to ease the length constraint, we apply the NP-chunks approach for selecting candidate terms. Furthermore, we remove a term that is subsumed by another term when determining the final candidate terms. For example, if there are two terms, Gaussian distribution and Gaussian distribution model, the first term will be filtered out.

After text pre-processing and candidate term selection, the next section will describe how to choose the discriminating features as input of our approach and which type of value to assign to these features. The approach taken to the term recognition is that of supervised machine learning, so the output of the machine learning algorithm is binary (a candidate term is either a term or not).

2.3 Feature Construction

Generally sentences at the beginning and the end of a paragraph are important, so a term occurring in the first sentence of a paragraph more likely to be valuable. For each candidate term, we are concerned not only with its frequency, but also with its specific position. As for term's position, we determine features that consist of eight elements and are divided into sentence level and location level. At the sentence level, we are concerned with two features, that is:

$\mathbf{P}_{cue}$: the candidate term and cue phrase are contained in a same sentence. We manually construct a cue phrase list according to our observations, e.g. *in this paper*, *in this article*, *we propose*, or *we use*. We check the candidate term whose affiliated words include any phrases of the cue phrase list. If a sentence covers both

candidate term and any phrases of cue phrase list, the $\mathbf{P}_{cue}$ value is assigned 1, otherwise 0.

$\mathbf{P}_{form}$: the candidate term is depicted in a specific form, e.g. uppercase, abbreviation or acronym. When capital letters appear in a term, this term may imply significant information in the context. Thus, concerning the value assignment of feature $\mathbf{P}_{form}$, the number of uppercase characters is assigned to it.

At the location level, we concentrate our attention on five features, that is:

$\mathbf{P}_{title}$: the candidate term is included in the title. Authors usually arrange the most important approach in the title to highlight their focus. If the term appears in the title section, the $\mathbf{P}_{title}$ value is assigned 1, otherwise 0.

$\mathbf{P}_{abs}$: the candidate term is included in the abstract section. An abstract can be regarded as the summary of a document since it always mentions the main methods and their purposes. If the term appears in the abstract section, the $\mathbf{P}_{abs}$ value is assigned 1, otherwise 0.

$\mathbf{P}_{ins}$: the candidate term is included in the introduction section. Since the introduction aims to introduce the outline of this article; it always discusses the major content including the target, applied method…etc. If the term appears in instruction section, the $\mathbf{P}_{ins}$ value is assigned 1, otherwise 0.

$\mathbf{P}_{exp}$: the candidate term is included in the experimental section. We observe that some techniques appear in the experimental section with low term frequency. For example, "Our system is developed in the Java platform." Nevertheless, the term Java appeared in this article only once. If we consider only the term frequency information, we would miss it. Hence, in order to ease the restriction of term frequency, we select the feature $\mathbf{P}_{exp}$ for keeping certain terms with low frequency. If the term appears in the experimental section, the $\mathbf{P}_{exp}$ value is assigned 1, otherwise 0.

$\mathbf{P}_{con}$: the candidate term is included in the conclusion section. Generally, the main method, the goal and experimental results are mentioned again in the last paragraph. If the term appears in conclusion section, the $\mathbf{P}_{con}$ value is assigned 1, otherwise 0.

In order to determine the above features, we develop a naïve method to divide a document into several blocks to accurately determine the range of each section.

$\mathbf{P}_{co}$: a candidate term co-occurs with associated words. The distribution of a word across lexical contexts is highly indicative of its meaning; moreover, some particular words appear together with other words and phrases. Therefore, to construct a co-occurrence list (i.e., associated terms) that has high association with technical terms, we adopt the logarithm of odds ratio (LODR) approach, as shown in Figure 1. We first obtain a sentence set containing the focused technical terms of training data. Then, the sentence set is split into two subsets of focused technical term set and other words set. Lastly, the LODR formula is used to generate associated terms that occur frequently together with a focused technical term, as follows:

$$LOD_r(t_i, F) = \log \frac{p/(1-p)}{q/(1-q)} = \log \frac{p(1-q)}{q(1-p)}$$

where, F is the focused technical term, t_i is any word in another words set. Let p be the probability that a word, t_i, co-occurs with any of the focused technical terms. That is,

p = Pr (t_i | sentence containing focused technical terms)

Also, let q be the probability that t_i co-occurs with non-focused technical terms. The formula for the occurrence probability of the word t_i with non-focused technical terms is shown below:

q = Pr (t_i | sentence excluding focused technical terms)

If LODr(t_i, F) > δ (where, δ is a threshold), then t_i is considered an associated term that focused technical terms co-occur with. The number of associated terms in the first three sentences S_k containing a candidate term c_i is assigned to feature P_{co}, as follows:

$$P_{co}(c_i, S_k) = \#\text{ of words}(S_k \cap \text{co-occurrence list})$$

where, $k = 3$.

In addition to the above features, we consider a term's within-document frequency, and the relative position of its first occurrence as well. Therefore, to this point we have considered ten features as input to the supervised learning algorithm for performing term recognition. After term recognition, we combine a strategy to perform the final term selection. We unify each term into a lowercase form to eliminate any duplicate terms.

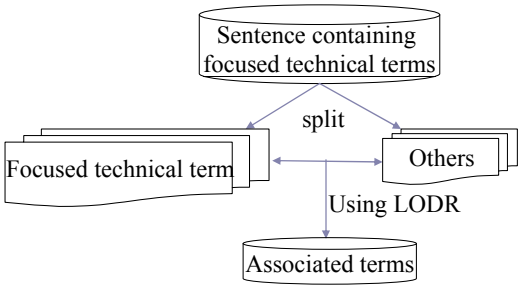

Figure 1. Process for generating associate terms.

3 Term Classification

In this section, we perform term classification to incorporate the results generated by term recognition. Given a focused technical term as input, we conduct three steps before the learning step: data collection, text pre-processing, and feature construction.

3.1 Data Collection and Text Pre-Processing

Due to the growth of the World Wide Web, many studies combine their proposed method with web-based resources. Sato and Sasaki [13] propose an automatic web-based method of collecting technical terms that are related to a given seed term. Turney [16] and Wong *et al*. [18] respectively apply the search engine and Wiki-pedia to calculate the mutual information between words. Thus, to implement term classification effectively, we regard each focused technical term as a seed term to collect related documents. Therefore, for each term, we rely on Google APIs [21] for gathering top-*k* (the default *k* is 3) related pages. Lastly, in order to deal with the context of each page more efficiently, we merge the top-*k* pages into a single document for the next procedure.

Due to the web page format, the retrieved pages contain many meaningless tags that are used for browser parsing, so we exact only the contexts which are contained in BODY tags. Stop words appear in text frequently and contain meaningless information. For example, "*is*", "*the*", and "*a*". Accordingly, we apply the stop-word list developed by Cornell University to remove the redundant words. The task of case folding involves converting each character of words into a specific form, such as upper case and lower case. In this case, we use the lowercase format as standard data representation.

3.2 Feature Construction

Since not all words are useful for vector representation, we select only words that are labeled with a noun tag, adjective tag or verb tag as features. Stemming is the fundamental text processing for reducing inflected words to their stem, we adopt the Porter's stemmer [11] since it has become the de-facto standard algorithm for English stemming. Lastly, we define a threshold to discard terms with low frequency.

In general, the most frequently used data representation in text mining is the bag-of-words. Thus, for our data representation, we use the vector space model (VSM), which is a way of representing documents through the words that they contain. Moreover, the value of each attribute (vector) can be assigned either a boolean value or a *tf-idf* (term frequency – inverse document frequency) value. Since the *tf-idf* weight is a statistical measure used to evaluate how important a word is to a document in a collection and the learning algorithm could distinguish this relatively accurately, we assign the *tf-idf* value to the entries of VSM. For technical trend analysis, statistics of term classification and time stamp information are needed, and this will be discussed in the experimental section (See Section 4.3).

4 Experimental and Results

The datasets utilized for experiments consists of 656 full-text papers from two well-known conferences, namely ACM SIGIR and KDD from 2002 to 2006. The experiments can be separated into three parts: term recognition, term classification, and technical trend analysis respectively for discussions.

4.1 Experiment of Term Recognition

We arbitrary select 100 papers from the total of 656 papers as the dataset for the first experiment since it is necessary to manually assignment of focused terms to each document. We then adopt the ModApte split to divide the data set into 75 documents (for training) and the remaining 25 documents (for testing). The average number of candidate terms for a document is 90 while the number of focused technical terms manually assigned to a document is 15. Each instance is assigned the feature-value pairs described in Section 2. Since the ratio of negative to positive terms can be very large, we conduct a sampling of negative terms to make the ratio approximately 1:1 in the training data set. We compare two supervised learning algorithms, namely SVM (support vector machine), and C4.5, for term recognition. These packages implemented in WeKa [22] software for the following tasks.

The precision (proportion of extracted terms that are marked as focused term to all the extracted terms), recall (proportion of the manually marked terms that are recognized, out of all marked terms available) and F-measure are used as the evaluation measures. The results for the 5-fold cross validation runs are shown in Table 1. As can be seen in this table, C4.5 gave 0.586 precision, 0.901 recall and 0.697 F-measure, whereas the SVM classifier gave better results, yielding 0.607 precision, 0.844 recall, and 0.706 F-measure. But neither SVM nor C4.5 can achieve good performance on precision. One of the reasons is the large ratio of negative to positive examples. Another possible reason is due to term variation. For example, authors may use *SVM approach* and *SVM method* to describe SVM technique in the context. Since domain experts rigidly subsume one term as a gold standard, the precision resulted in low performance.

Although there is no existing work explicitly concentrates on the task of focused technical term recognition, they are some researches on automatic keywords extraction. Hence, we compare our term recognition with two keyword-based methods. The first one is called Keyword-Field, which takes the

terms in keyword section (or terms appear in titles when keyword section is not available) as the focused technical terms. The second one is Hulth's proposal in [2], which improves automatic keyword extraction through linguistic knowledge. We implement the proposed method with corresponding feature-value pair suggested in the paper to extract keywords from the same dataset as mentioned above. As we can see Keyword-Field has moderate precision (0.448), but low recall (0.199), which yields 0.267 F-measure. One explanation of the disappointing recall is the limited number of keywords assigned by authors. However, the keywords provided by authors not only contains focused technical terms, but also other concept terms. As for the Hulth's method, it yielded much lower precision (0.378) with 0.459 F-measure. Since the selections of feature-value are devised for the task of keyword extraction, accordingly our performance can outperform theirs.

Table 1. Results of term recognition.

Approach	# of Assigned terms	Precision	Recall	F-measure
C4.5	15	0.568	0.901	0.697
SVM	**15**	**0.607**	**0.844**	**0.706**
Keyword Field	4	0.448	0.199	0.276
Hulth's	15	0.378	0.584	0.459

4.2 Experiment of Term Classification

For the term classification experiment, we use the result from term recognition in all the documents (656 papers) as the input. After automatic term recognition processing, 8624 terms were extracted. The goal of this experiment is to classify these terms into a suitable technology category according to their meaning. We adopt ACM's computing classification system to select five technology categories: Artificial Intelligence (AI), Pattern Recognition (PR), Probability and Statistics (PS), Information Storage and Retrieval (IR) and Database Management (DBM). Each term is manually classified into either at most one category or discarded. For example, we classify "language model" and "text analysis" as AI since NLP is a subcategory under AI. In detail, AI, PR, PS, IR, and DBM contain 891, 750, 877, 1428 and 875 technical terms, respectively. In other words, domain experts discarded 3803 terms that were not related to these categories.

We use the SVM classifier with the *tf-idf* value and conduct four-fold cross-validation in the following experiments. The first part of the experiments is to verify whether Web resources can improve the classification accuracy. For each focused technical term, we use the first three sentences containing the word to extract the features mentioned in Section 3.2. However, we already see a great difference between the original research paper (39.10 in terms of precision) and our Web resource (71.21). We then expand the contexts of these three sentences (one sentence before and after the sentences) to include further input. For Web resources, we have an increase to 74.63, but a decrease to 30.00 in original research paper. Finally, all the sentences including the focused technical terms are used such that all *K* pages have an influence in the experiments. Again, we have an increase to 77.87 using Web resource, but only 36.96 from the original research paper. Overall, the performance using a Web resource is better than that using the original research papers. The values for precision, recall, and F-measure are shown in Table 2.

Table 2. Results for term classification on different volumes of data.

Data Source	Volume of Data	Precision (%)	Recall (%)	F-score (%)
Web Page	Top-three sentences	71.21	45.35	55.41
	Top-three-sentences and its context	74.63	43.78	55.19
	All sentences	77.87	44.20	56.40
Research paper	Top-three sentences	39.10	28.60	33.03
	Top-three-sentences and its context	30.00	23.59	26.41
	All sentences	36.96	31.14	33.80

The results indicate that web resources and the original research papers can achieve F-measures of around 56.40% and 33.80, respectively. Clearly, with the assistance of Web resources, better performance can be achieved than by using the original research papers alone. Since there probably exists an explicit definition for a particular term on the web, more discriminative features can be found in classification. On the contrary, since an original research paper generally focuses on its research topic, not a specific term, thus the sentences containing the term may include some noise. From this, it seems reasonable to conclude that using the web can collect more related information on a particular term to enhance performance.

The experiment that we consider next is how different types of POS features can affect classification

performance. We use all sentences containing the focused technical terms as input; and then select only three syntactical types, namely adjective, noun and verb, as our feature words. The values for precision, recall, and F-measure are shown in Table 3.

Again, the web page resource could outperform the original research papers. Interestingly, both adjective words and verb words were unable to improve classification performance; and the F-measures have been dropped to around 47.24% and 50.08 for Web resources, respectively. However, noun words provided a significant improvement in terms of F-measure (from 56.40 to 64.37) in Web resources. That is because nouns are more relevant from an applicative viewpoint since they comprise a large portion of the document. In addition, nouns can conceptually match the focused technical terms (which are also nouns), while other words (e.g., verbs, adjectives, or adverb) tend to be more functional and general. From what has been discussed above, we can conclude that the combined used of a web resource can easily collect information on a specific keyword and that nouns can provide more discriminating power for improving term classification.

Table 3. Results for term classification on different POS tags.

Data Source	POS feature	Precision (%)	Recall (%)	F-score (%)
Web Pages (All sentences)	Adjective	56.00	40.84	47.24
	Noun	**76.67**	**55.47**	**64.37**
	Verb	57.84	44.17	50.09
	Combined (Adj.+N+V)	74.85	47.70	58.25
Research papers (All sentences)	Adjective	25.61	19.29	20.00
	Noun	**33.77**	**34.03**	**33.39**
	Verb	20.38	18.08	19.16
	Combined (Adj.+N+V)	20.58	40.37	24.65

4.3 Experiment of Technical trend analysis

We integrate the results of term classification with temporal information to demonstrate the evolutionary technical trend in variant research topics using visual presentations. We select the best results of term classification that are generated by the Web resource and regard conference names as our research topics, that is, SIGIR and KDD and use the publication dates as time-stamps. For a certain research topic in a particular year, the strength of a technology category is computed by the portion of focused technical terms in that year. For example, suppose we have 1000 focused technical terms in 2005 on SIGIR and 200 technical terms are classified into AI category. Then, the strength of AI in 2005 for SIGIR is 0.2.

Figure 2 and Figure 3 illustrate the evolutionary technical trend in these two research conferences, respectively. For each figure, five different color areas depict five technology categories, as mentioned in Section 4.2. As we can see, in KDD conferences, the strengths of DBM and PS techniques are higher than other techniques. That is because KDD mainly concentrates on knowledge discovery and data mining, which combine with much more probability model. With the SIGIR research topic, the strengths of IR and AI technique are higher than other techniques since SIGIR focuses principally on information retrieval systems and natural language processing. In both figures, we can notice some slight bursts on PS technique from 2004 to 2005. Indeed, PS has become the major technique used in addition to database management (in KDD) and information retrieval (in SIGIR), especially in recent years. The different focuses of KDD and SIGIR on DBM and IR techniques, respectively, can also be observed clearly from our technical trend figure. Overall, these results roughly corresponded to contemporary research trends from the well-known conferences we selected.

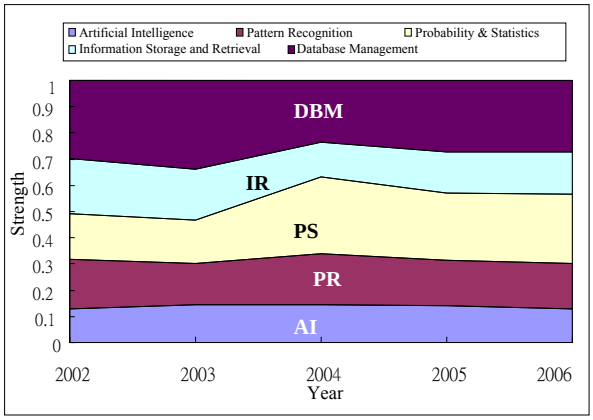

Figure 2. Technical trends of KDD.

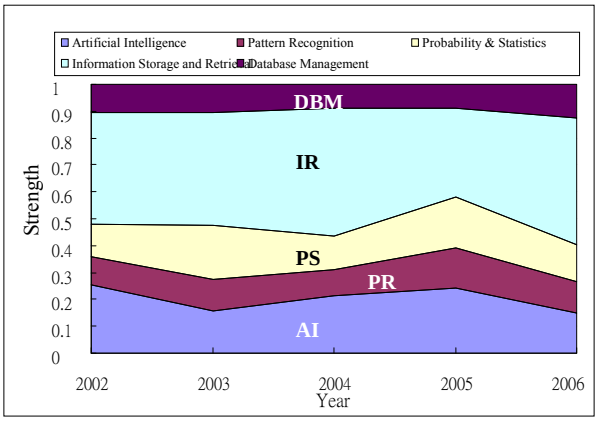

Figure 3. Technical trends of SIGIR.

To investigate the most often used techniques for each technology category; we list focused technical terms with high frequency in alphabetical order and depict them in different font sizes according to their frequencies (see Table 4). We have seen "language model", "vector space model", "singular value decomposition", "tdt2 corpus" as the focused technical terms used in AI since it covers the text processing domain, including *language parsing*, *knowledge representation*, etc.. For the pattern recognition domain, "KL divergence retrieval model", "maximum entropy model" and "smoothing methods" are the most used terms. In the probability and statistics domain, we found "dirichlet distribution" has become as important as "gaussian distribution" and "mixture model". As for information retrieval and storage domain, not only did we found the commonly used IR systems such as Okapi and Smart, we also found that many studies have conducted experiments on web resources (e.g., msn web corpus, and internet movie database) in addition to TREC collection. Finally, in the domain of database management, many data mining algorithms (e.g., SVM, decision trees, Bayesian network model, EM) can all be discovered as expected. Overall, we can easily comprehend basic techniques and detect the technique used in each category from this table.

Table 4. Popular terms for each technology category.

Categories	Popular techniques
Artificial Intelligence	bootstrap method, **language model**, **language modeling approach**, light stemmer, lsi method, named entity recognizer, principal component analysis, **singular value decomposition**, **tdt2 corpus**, **vector space model**
Pattern Recognition	cosine similarity, **distance function**. **kernel method**, information gain, **js divergence**, **kl divergence retrieval model**, **maximum entropy model**, pyramid method, rbf kernel, **smoothing method**
Probability and Statistics	binomial distribution, **dirichlet distribution**, **gaussian distribution**, **hidden markov model**, hmm, **markov model**, **mixture model**, **normal distribution**, poisson distribution, **uniform distribution**, **t test**
Information Storage and Retrieval	document object model, **html tag,** internet movie database, **ir system**, msn web corpus, ohsumed collection, **okapi retrieval model**, pagerank citation ranking, ranking method, **smart retrieval system**, search engine, **tf idf**, **trec collection**, **web page**, webkd dataset, wt10g collection
Database Management	association rule, bayesian network model, **decision tree**, **expectation maximization**, frequent itemset, **information bottleneck method**, knn, rule based classifier, sequential pattern, **support vector machine**, **svm classifier**,

5. Related works

Several studies have been proposed to automatically extract scientific or technical terms from domain-specific corpora [7][8][17]. A standard automatic term recognition process consists of two procedures. Extracting candidate terms from corpora occurs in the first procedure and assign a weight (score) to each candidate term in the second.

Candidate term extraction: A candidate term is usually built in a form of single word or multi-word. To extract compound words as a potential candidate term, three term selection approaches have been widely used: *n*-grams, noun phrase chunks and part-of-speech tag patterns. Zhai and Evans [1] consider noun phrases, Wermter and Hahn [16] consider *n*-grams, while Frantzi *et al*. [1] define three filters that consist of pos tag patterns for covering all potential terms. In ever view, Hulth [2] compares the above three methods and the experimental results indicate that noun phrase chunks obtain the best performance for precision.

Weighting function computes scores indicating the degree to which a candidate term is a terminological unit. Two representative methods have been proposed, the C-value/NC-value approach [2] and the limited paradigmatic modifiability approach [17]. The purpose of these methods is to improve the extraction of nested terms.

In supervised automatic keyword extraction, a classifier is trained by using documents with annotated keywords. The training model is subsequently applied to test documents for determining each term from these documents as either a keyword or a non-keyword. The task, applying supervised machine learning for automatic keyword extraction, was first proposed by Turney [15]. Features used for the learning task include: the relative number of characters of the phrases; the first relative occurrence of a phrase component; and whether the last word is an adjective. Hulth in [3][4] improves automatic keyword extraction by adding linguistic knowledge to the feature representation. Another difference between these works is that Turney conducts experiments on full-length text and restricts the length of keyword to be within three tokens, whereas Hulth adopts an arbitrary length of keyword from abstract of journal paper as corpus. Although we similarly apply the supervised machine learning approach for term recognition, there are some differences between these studies and ours. First, our goal is to recognize the focused technical terms. Thus, the feature selection and feature value assignment are quite distinct since not all of the keywords are technical terms. For example, for each candidate term, we are not only concerned with its

frequency, but also its specific position. Second, we also use chunk phrases as candidate terms, as does Hulth, but we choose the full-text conference papers to retain the completeness of information and integrity of context for determining features.

Term classification has been learned by typical classifiers, such as the hidden markov model, naïve bayes, decision tree, support vector machine, etc. Whether the term frequency within documents of the specific domain is higher than its frequency within generic documents is the main feature for such classification task. Nenadic *et al*. [9] conducted a series of large-scale experiments with different types of features for a multi-class SVM. The features used include, document identifiers, single words, their lemmas and stems. Spasic *et al*. in 2003 [14] suggest the use of a genetic algorithm as a learning engine, where the verb complementation patterns are automatically learned by combining information found in a corpus and ontology. In our approach, we used SVM as the learning techniques and added features from the Web resources for term classification.

6 Conclusions

In this paper, we propose the task of exploring applied techniques in academic research papers and use these focused technical terms for technical trends analysis. Our proposed frameworks first recognize the focused technical terms and then combine web resources to conduct term classification. Next, we use the strength of each technology category in each time period to find the evolutionary technical trends through visual representations. This information allows researchers to grasp the trends of technology strength and the latent popular technical terms in each research topic.

We evaluate our proposed approaches using full-text ACM conferences papers from 2002 to 2006. For the extraction of focused technical terms, the proposed approach can achieve 60.8%, 84.4%, and 70.6, for precision, recall, and F-measure, respectively. For term classification, the approach can achieve 76.67%, 55.47% and 64.37% for precision, recall, and F-measure, respectively. By using the focused technical terms, researchers can easily identify what technique has been used in a paper and to which category it belongs via term classification. In addition, we can better understand contemporary research trends and techniques through the technology strength figure and most frequent terms in each category.

For future work, classification of papers by the problems that they solve will be helpful for users to better understand the techniques or algorithms used in specific problems.

7. References

[1] D.A. Evans, and C. Zhai, "Noun-phrase analysis in unrestricted text for information retrieval," In 34th ACL, ACM Press, New York, 1996, pp. 17-24.

[2] K. Frantzi, S. Ananiadou, and H. Mimaz, "Automatic recognition of multi-word terms: the C-value/NC-value method," J. Digital Libraries, Vol. 3, 2000, pp. 115-130.

[3] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," In Empirical Methods in Natural Language Processing, 2003, pp. 216-223.

[4] A. Hulth, B.B. Megyesi, "A Study on Automatically Extracted Keywords in Text Categorization," In 44th ACL, ACM Press, New York, 2006, pp. 537-544.

[5] A. Kontostathis, L.M. Galitsky, W.M. Pottenger, S. Roy, and D.J. Phelps, "A survey of emerging trend detection in textual data mining. Springer Verlag, London, 2003.

[6] Q. Mei, and C. Zhai, "Discovering evolutionary theme patterns from text: an exploration of temporal text mining," In 11th SIGKDD, ACM Press, New York, 2005, pp. 198-207.

[7] H. Nakagawa, "Automatic Term Recognition based on Statistics of Compound Nouns," J. Terminology, Vol. 6, 2000, pp. 195-210.

[8] G. Nenadić, S. Ananiadou, and J. McNaught, "Enhancing automatic term recognition through recognition of variation," In 20th COLING, ACM Press, New York, 2004, pp. 604-610.

[9] G. Nenadić, I. Spasić, and S. Ananiadou, "Automatic discovery of term similarities using pattern mining," In 18th COLING, ACM Press, New York, 2002, pp. 1-7.

[10] J. Perki¨o, W. Buntine, and S. Perttu, "Exploring Independent Trends in a Topic-Based Search Engine," In International Conference on the Web Intelligence, IEEE Press, New York, 2004, pp. 664-668.

[11] M.F. Porter, "An algorithm for suffix stripping," J. Program: electronic library & information systems. 40, 2006, pp.211-218.

[12] K. Rajaraman, and A.H., Tan, "Topic detection, tracking, and trend analysis using self-organizing neural networks," In 5th PAKDD, Spring-Verlag Press, London, 2001, pp. 10-107.

[13] S. Sato, and Y. Sasaki, "Automatic collection of related terms from the web," In 41st ACL, ACM Press, New York, 2003, pp. 121-124.

[14] I. Spasić, G. Nenadić, and S. Ananiadou, "Using Domain-Specific Verbs for Term Classification," In Proc. of the ACL 2003 workshop on Natural language processing, 2003, pp. 17-24.

[15] P.D. Turney, "Learning Algorithms for Keyphrase Extraction," J. Information Retrieval. 2, 2000, pp. 303-336.

[16] P.D. Turney, "Mining the Web for synonyms: PMI-IR versus LSA on TOEFL," In 12th ECML, Spring-Verlag Press, London, 2001, pp. 491-502.

[17] J. Wermter, and U. Hahn, "Finding new terminology in very large corpora," In 3rd international conference on Knowledge capture, ACM Press, 2005, pp. 137-144.

[18] W. Wong, W. Liu, and M. Bennamoun, "Featureless similarities for terms clustering using tree-traversing ants," In Proc. of the 2006 international symposium on Practical cognitive agents and robots, 2006, pp. 177-191.

[19] Harvard University: http://www.wjh.harvard.edu/~inquier/

[20] Princeton University: http://wordnet.prnceton.edu

[21] Google Code: http://code.google.com/

[22] Waikato University: http://www.cs.waikato.ac.nz/ml/weka/