To Improve the Effectiveness of Discovered Patterns for Text Mining

Priyanka Chandragiri[1], C.Prathusha[2], K.Bhanu Prasad[3]
[1]Department of Computer Science Engineering, Christu jyothi Institute of Technology and Science, Jangoan, Warangal, India.
[2]Department of Information Technology, Vaagdevi Engineering College, Bollikunta, Warangal, India.
[3]Department of Computer Science Engineering, Vaagdevi Engineering College, Bollikunta, Warangal, India.

‘

ABSTRACT

A significant number of data mining techniques have been presented in order to perform different knowledge tasks. These techniques include association rule mining, frequent item-set mining, sequential pattern mining, maximum pattern mining, and closed pattern mining. These data mining techniques have been proposed for mining useful patterns in text documents. Most existing text mining methods are adopted term-based approaches; this does not solve the problem. The phrase-based approaches includes the process of pattern deploying and pattern evolving, as this approach arise a problem of low frequency and misinterpretation, an IPE algorithm is used. In order to improve the effectiveness of using and updating discovered patterns according to users view, experiments on RCV1 data collection and TREC topics achieves user required data from a large document by removing noisy pattern by pattern evolving and splitting the document according to the Data Base Management .

KEYWORDS: *Text organization, Pattern mining, Pattern surfacing, Information filter, Pattern Deploying, Pattern Evolution.*

I. INTRODUCTION

Text mining is the technique that helps users find useful information from a large amount of digital text documents on the Web or databases. It is therefore crucial that a good text mining model should retrieve the information that meets users’ needs within a relatively efficient time frame. Traditional Information Retrieval (IR) has the same goal of automatically retrieving relevant documents as many as possible while filtering out non-relevant ones at the same time [3]. However, IR-based systems cannot meet users’ needs [9]. Many text mining methods have been developed in order to achieve the goal of retrieving useful information for users [2]. Most of them adopt the keyword-based approach, whereas the others choose the phrase-based technique to construct a text representative for a set of documents. Although phrases contain less ambiguous and narrower meanings than individual words, the likely reasons for the discouraging performance from the use of phrases are:(1) phrases have inferior statistical properties to words, (2) they have low frequency of occurrence, and (3) there are a large number of redundant and noisy phrases among them [5]. D.D Lewis [1] also suggested that simple phrase-based representations are not worth to pursue since they found no significant performance improvement on eight different representations based on words, phrases, synonyms and Hyponyms.

In this paper we propose two novel pattern deploying algorithms to effectively exploit discovered patterns for the text mining problem.

II. RELATED WORKS

In [9], term frequency and inverse document frequency (TFIDF) weighting scheme is used for text representation. In addition to TFIDF, the global idf and entropy weighting scheme is proposed by [3] and improves the performance by an average of 30%. The problem of the bag of words approach is how to select a limited number of features among an enormous set of words or terms in order to increase the system's efficiency and avoid the over fitting [6]. To reduce the number of features, many dimensionality reduction approaches have been conducted by the use of feature selection techniques, such as Information Gain, Mutual Information, Chi-Square, Odds ratio, and so on. The details of these selection functions are stated in [6].

The choice of a representation depends on what one regards as the meaningful units of text and the meaningful natural language rules for the combination of these units [5]. With respect to the representation of the content of documents, some research works have used phrases rather than individual words propose a phrase-based text representation for web document management using rule-based Natural Language Processing (NLP) and Context Free Grammar (CFG) techniques. By apply data mining techniques to text analysis by extracting co-occurring terms as descriptive phrases from document collections.

2.1 text representation

2.1.1 Keyword-based Representation

The bag-of-words scheme is a typical keyword-based representation in the area of information retrieval. It has been widely used in text classification tasks due to its simplicity. Figure.1 illustrates the paradigm of the bag-of-words technique. As we can see that each word in the document is retrieved and stored in a vector space alone with its frequency. The context of this document then can be represented by these words, known as "features". However, the main drawback of this scheme that the relationship among words cannot be reflected[2]. Another problem in considering single words as features is the semantic ambiguity which can be categorized in:

- Synonyms: a word which shares the same meaning as another word (e.g. taxi and cab)
- Homonym: a word which has two or more meanings (e.g. river "bank" and CITI "bank").

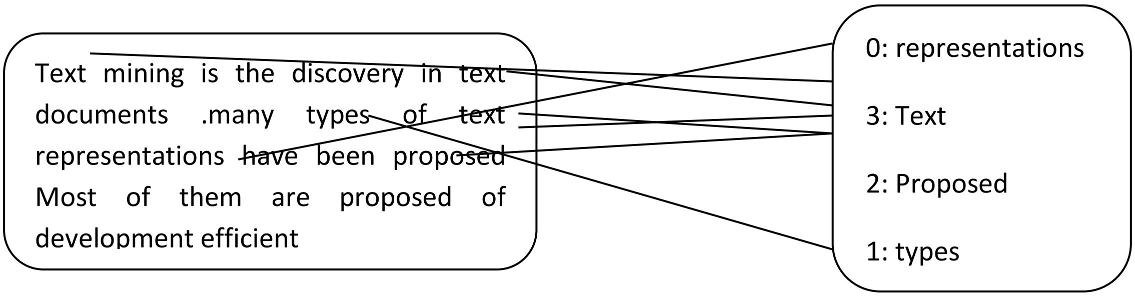

Text mining is the discovery in text documents .many types of text representations have been proposed Most of them are proposed of development efficient
0: representations
3: Text
2: Proposed
1: types

Figure 1: Bag of words Representation

1.2 Phrase-based Representation

Using single words in keyword-based representation poses the semantic ambiguity problem. To solve this problem, the use of multiple words (i.e. phrases) as features using phrase based representation is proposed [10]. In general, phrases carry more specific content than single words. For instance, "engine" and "search engine" .To identify groups of words that create meaningful phrases is a better method, especially for phrases indicating important concepts in the text. Lewis [4] noted that the traditional term clustering methods are unlikely to provide improved text representation.

III. PATTERN TAXONOMY MODEL

In PTM, we split a text document into a set of paragraphs and treat each paragraph as an individual transaction, which consists of a set of words (terms). At the subsequent phase, we apply the data mining method to find frequent patterns from these transactions and generate pattern taxonomies.

Two main stages are considered in PTM. The first stage is how to extract useful phrases from text documents, which will be discussed in this chapter. The second stage is then how to use these discovered patterns to improve the effectiveness of a knowledge discovery system [11].

Patterns can be structured into taxonomy by using the subset relation. Smaller patterns in the taxonomy are usually more general because they could be used frequently in both positive and negative documents and larger patterns are usually more specific since they may be used only in positive documents. The semantic information will be used in the pattern taxonomy to improve the performance of using closed patterns in text mining.

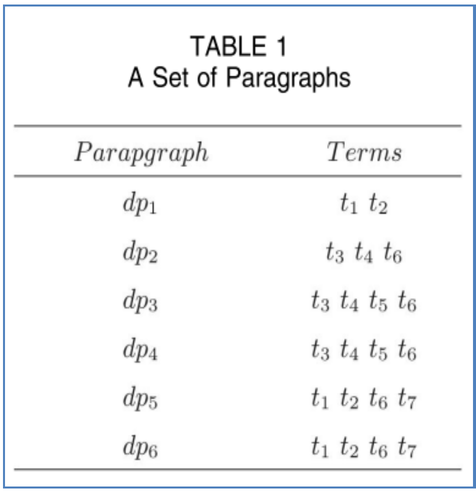


TABLE 1
A Set of Paragraphs

Parapgraph	*Terms*
dp_1	t_1 t_2
dp_2	t_3 t_4 t_6
dp_3	t_3 t_4 t_5 t_6
dp_4	t_3 t_4 t_5 t_6
dp_5	t_1 t_2 t_6 t_7
dp_6	t_1 t_2 t_6 t_7

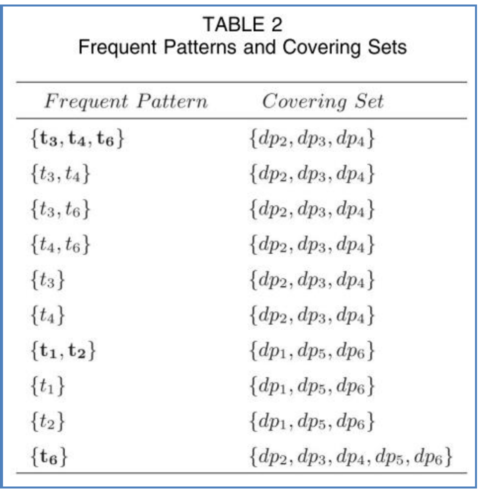


TABLE 2
Frequent Patterns and Covering Sets

Frequent Pattern	*Covering Set*
$\{\mathbf{t_3, t_4, t_6}\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3, t_4\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3, t_6\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_4, t_6\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_4\}$	$\{dp_2, dp_3, dp_4\}$
$\{\mathbf{t_1, t_2}\}$	$\{dp_1, dp_5, dp_6\}$
$\{t_1\}$	$\{dp_1, dp_5, dp_6\}$
$\{t_2\}$	$\{dp_1, dp_5, dp_6\}$
$\{\mathbf{t_6}\}$	$\{dp_2, dp_3, dp_4, dp_5, dp_6\}$

Table1: shows a set of Paragraphs, **Table 2:** shows frequent patterns and covering sets.

3.1 Finding Frequent and Closed Sequential Pattern

When feature selection process is completed, the frequent and closed patterns are discovered based on the documents, the termset ‘X’ in document ‘d’, $[x]$ is used to denote the covering set of X for d, which includes all paragraph ‘dp ’$\in$ PS(d) [5] such that X $\subseteq$dp . i.e X = $\{ dp \mid dp \in ps(d), x \subseteq$dp} Its absolute support is the number of occurrences of X in PS (d) that is, sup$_d$ (X) = $|\,|[x]|\,|$ its relative support is the fraction of paragraphs that contain the pattern, i.e

$$\mathrm{Sup}_d\,(X) = \frac{|[x]|}{|ps(d)|}$$

A sequential pattern X is called frequent pattern if its relative support (or absolute support) is a minimum support. Some property of closed patterns can be used to define closed sequential patterns. The algorithm for finding the support count is given in Sequential Pattern Mining (SPM).

3.1.1 Sequential Pattern Mining (SPM)

The basic definitions of sequences used in this research work are described as follows. Let T = {t1, t2, . . . , tk } be a set of all terms, which can be viewed as words or keywords in text documents. A sequence S = s1 , s2 , . . . , sn (si∈ T) is an ordered list of terms. Note that the duplication of terms is allowed in a sequence. This is different from the usual definition where a pattern consists of distinct terms which is defined in algorithm 1[11].

An example of sub-sequence: A sequence α = a1 , a2 , . . . , an is a sub- sequence of another sequence β = b1 , b2 , . . . , bm , denoted by α β there exist integers 1 ≤ i1 < i2 < . . . < in ≤ m, such that a1 = bi1 , a2 = bi2 , . . . , an =bin .

Algorithm 1 :

SPMining(*PL*, *min_sup*)

Input: the list of *nTerms* frequent sequential patterns *PL*; The minimum support threshold *min_sup*. (Notice: in the beginning, *SP* is the set of *1Terms* frequent sequential patterns.)

Output: a set of frequent sequential patterns *SP*.

Method:

```
SP = SP – {$P_a$   SP |   $P_b$   PL such that
     len ($P_a$) = len( $P_b$) -1 ∧ $P_a$   $P_b$ ∧
     $supp_a$($P_a$) = $supp_a$($P_b$) }  /* pruning */
SP   SP   PL   /* add found patterns */
PL′   { }   /* PL′ : set of (n+1)Terms
   frequent sequential patterns */
foreach pattern p in PL do begin
    generate p-projected database PD
    foreach frequent term t in PD do begin
        P′   p   t   /* P′ : set of (n+1)Terms
                        sequential candidates */
        if $supp_r$(P′ ) ≥ min_sup then
            PL′   PL′   P′
        end if
    end for
end for
if |PL′ | = 0 then
    return /* no more patterns found */
else
    call SPMining(PL′, min_sup)
end if
```

SPMining adopts the concept of projected database method for extracting frequent sequential patterns from a document. The main difference between SPMining and others [17] [23], which adopt the same concept, is that SPMining deals with several sequences at a time, whereas others only handle one sequence at a time.

3.2 Feature Selection Method

In this methods documents are considered as inputs and the features for the set of documents are collected, features are collected based on TFIDF. However many terms with larger weights (e.g term frequency and inverse document frequency (tf*idf) weighting schemes) are general terms because they can be frequently used in both relevant and irrelevant information. The feature selection approach is used to improve the accuracy of evaluating term weights because the discovered patterns are more specific then the whole documents. In order to reduce the irrelevant features in dimensionality reduction approaches, the use of features selection technique is adopted.

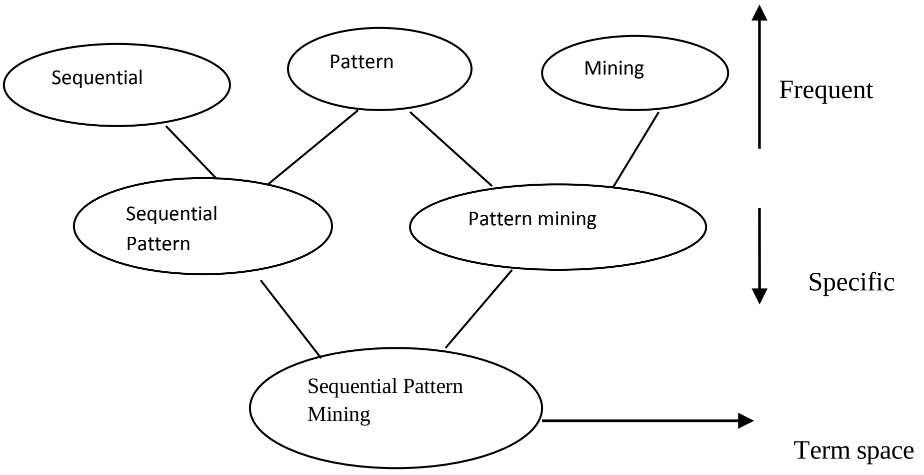


Figure 2: Pattern Pruning

IV. PATTERN DEPLOYING METHOD

In order to use the semantic information in the pattern taxonomy to improve the performance of closed patterns in text mining, we need to interpret discovered patterns by summarizing them as d-patterns in order to accurately evaluate term weights (supports).Terms are weighted according to their appearances in discovered closed patterns.

4.1 Representations of Closed Patterns

It is complicated to derive a method to apply discovered patterns in text documents for information filtering system.

$$P_1 \oplus P_2 = \{(t,x_1 + x_2 |(t,x_1)\} \in p_1,(t,x_2) \in p_2\} \cup \{(t,x)|(t,x) \in p_1 \cup p_2, \text{not}(t, \_) \in P_1 \cap P_2)\}$$

where _ is the wild card that matches any number.
For the special case we have $p \oplus \emptyset = p$; and the operands of the composition operation are interchangeable. The result of the composition is still a set of term-number pairs [8].

$$\widehat{d1} = \{(t_{i1}, n_{i1}),(t_{i2} , n_{i2}),\ldots\ldots(t_{im}, n_{im})\}$$

Let DP be a set of d-patterns in D^+ and $p \in$ DP be a d-pattern. We call p(t) the absolute support of term t, which is the number of patterns that contain t in the corresponding patterns taxonomies. In order to effectively deploy patterns in different taxonomies from the different positive documents, d-patterns will be normalized using the following assignment sentence:

$$p(t) \longleftarrow p(t) \text{ X } \frac{1}{\sum_{t \in T} P(t)}$$

for all $p_i \in$ DP, where

p_i = f{(t1, f1); (t2, f2); . . . ; (tk, fk)} 2 DP;

$$wi = \frac{f_i}{\sum_{j=1}^{k} f_j}$$

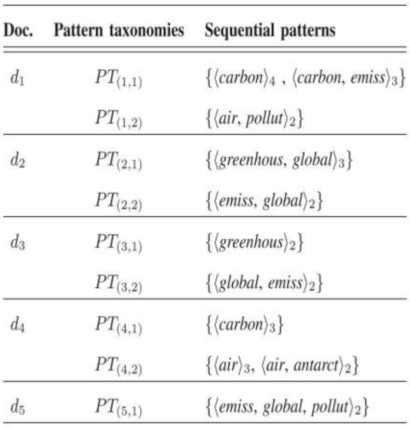
and T = {t|(t,f) $\in$ p,p $\in$ DP}..

β(p $_I$) is called the normal form (or normalized d-pattern) of d-pattern p_i in this paper, and termset(p_i) = {t_1, t_2,…t_k}.

4.2 D-Pattern Mining Algorithm

To improve the efficiency of the pattern taxonomy mining, an algorithm, SP Mining, was proposed in [5] to find all closed sequential patterns, which used the well-known Apriori property in order to reduce the searching

d1 = {(carbon; 2); (emiss; 1); (air; 1); (pollut; 1)};
d2 = {(greenhous; 1); (global; 2); (emiss; 1)};
d3 = {(greenhous; 1); (global; 1); (emiss; 1)};
d4 = {(carbon; 1); (air; 2); (antarct; 1)};
d5 = {(emiss; 1); (global; 1); (pollut; 1)}:



Doc.	Pattern taxonomies	Sequential patterns
d_1	$PT_{(1,1)}$	$\{\langle carbon\rangle_4\ ,\ \langle carbon, emiss\rangle_3\}$
	$PT_{(1,2)}$	$\{\langle air, pollut\rangle_2\}$
d_2	$PT_{(2,1)}$	$\{\langle greenhous, global\rangle_3\}$
	$PT_{(2,2)}$	$\{\langle emiss, global\rangle_2\}$
d_3	$PT_{(3,1)}$	$\{\langle greenhous\rangle_2\}$
	$PT_{(3,2)}$	$\{\langle global, emiss\rangle_2\}$
d_4	$PT_{(4,1)}$	$\{\langle carbon\rangle_3\}$
	$PT_{(4,2)}$	$\{\langle air\rangle_3, \langle air, antarct\rangle_2\}$
d_5	$PT_{(5,1)}$	$\{\langle emiss, global, pollut\rangle_2\}$

Table 3: Example of a Set of Positive Documents Consisting of Pattern Taxonomies

Let m =| T | be the number of terms in T , n | D^+| be the number of positive documents in a training set, K be the average number of discovered patterns in a positive document, and k be the average number of terms in a discovered pattern. We also assume that the basic operation is a comparison between two terms.

The time complexity of the d-pattern discovery (from steps 6 to 9) is $O(Kk^2 n)$. Step 10 takes O(mn) Step 12 also gets all terms from d-patterns and takes O ($m^2 n^2$). Steps 13 to 15 initialize support function and take O(m),, and the steps 16 to 20 take O(mn). Therefore, the time complexity of pattern deploying is

$$O(Kk^2 n + mn + m^2 n^2 + m\, \text{þ} + mn) = O(Kk^2 n + m^2 n^2)$$

After the supports of terms have been computed from the training set, the following weight will be assigned to all incoming documents d for deciding its relevance

Weight (d) $=\sum_{t\in T} support\,(t)\ \tau(t,d)$

Where $\tau(t,d) = 1$ if $t \in d$;

Otherwise $\tau(t,d) = 0$

V. INNER PATTERN EVOLUTION

This section, we discuss how to reshuffle supports of terms within normal forms of d-patterns based on negative documents in the training set. The technique will be useful to reduce the side effects of noisy patterns because of the low-frequency problem. This technique is called inner pattern evolution here, because it only changes a pattern's term supports within the pattern.

A threshold is usually used to classify documents into relevant or irrelevant categories. Using the d-patterns, the threshold can be defined naturally as follows:

$$\text{Threshold (DP)} \ = \ \min_{p\in DP} \Big(\sum_{(t,w)\in\beta(p)} \sup port(t) \Big)$$

A noise negative document nd in D^- is a negative document that the system falsely identified as a positive, that is weight (nd) ≥ Threshold (DP) . In order to reduce the noise, we need to track which d-patterns have been used to give rise to such an error. We call these patterns offenders of nd.(offender)

An offender of nd is a d-pattern that has at least one term in nd

The set of offenders of nd is defined by:

$\Delta(nd) = \{p \in DP \mid \text{terms set}(p) \cap nd \neq \emptyset\}$

There are two types of offenders:

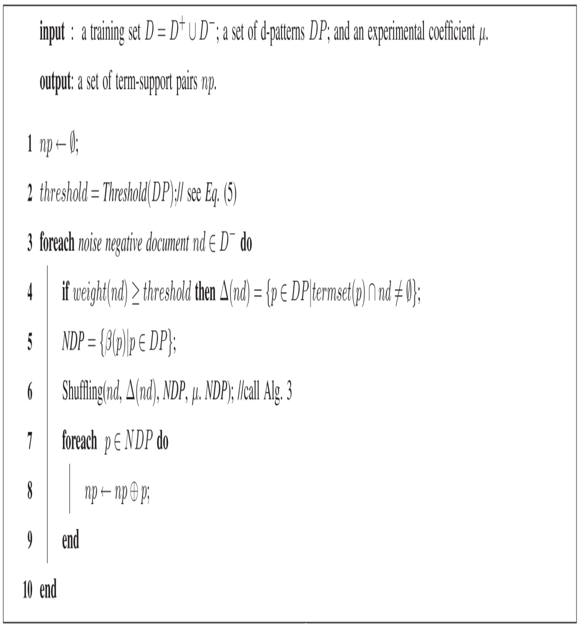
1) A complete conflict offender which is a subset of nd; and

2) A partial conflict offender which contains part of terms of nd.

The main process of inner pattern evolution is implemented by the algorithm IPEvolving (see Algorithm 2). The inputs of this algorithm are a set of d-patterns DP, a training set D = D^+

U D. Here term supports are defined by using all noise negative documents. Which is used to find noise documents and the corresponding offenders and gets normal forms of d-patterns NDP.

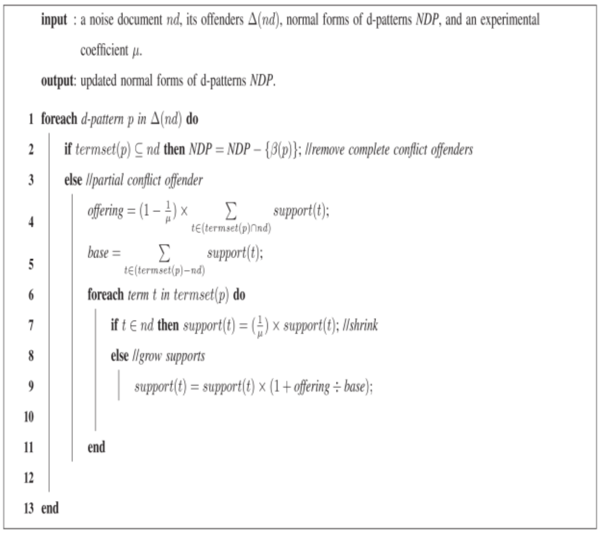
For example, for the following d-pattern
$\hat{d}=\{(t_1,3),(t_2,3,(t_3,3),(t_4,3),(t_6,8)\}$



```
input : a training set $D = D^+ \cup D^-$; a set of d-patterns $DP$; and an experimental coefficient $\mu$.
output: a set of term-support pairs $np$.

$np \leftarrow \emptyset$;
$threshold = Threshold(DP)$;// see Eq. (5)
foreach noise negative document $nd \in D^-$ do
    if $weight(nd) \geq threshold$ then $\Delta(nd) = \{p \in DP | termset(p) \cap nd \neq \emptyset\}$;
    $NDP = \{\beta(p) | p \in DP\}$;
    Shuffling($nd$, $\Delta(nd)$, $NDP$, $\mu$. $NDP$); //call Alg. 3
    foreach $p \in NDP$ do
        $np \leftarrow np \oplus p$;
    end
end
```

Figure 3 : Algorithm 2:IPE Evolving(D^+,D^-,DP,μ).



```
input : a noise document $nd$, its offenders $\Delta(nd)$, normal forms of d-patterns $NDP$, and an experimental
        coefficient $\mu$.
output: updated normal forms of d-patterns $NDP$.

foreach d-pattern $p$ in $\Delta(nd)$ do
    if $termset(p) \subseteq nd$ then $NDP = NDP - \{\beta(p)\}$; //remove complete conflict offenders
    else //partial conflict offender
        $offering = (1 - \frac{1}{\mu}) \times \sum_{t \in (termset(p) \cap nd)} support(t)$;
        $base = \sum_{t \in (termset(p) - nd)} support(t)$;
        foreach term $t$ in $termset(p)$ do
            if $t \in nd$ then $support(t) = (\frac{1}{\mu}) \times support(t)$; //shrink
            else //grow supports
                $support(t) = support(t) \times (1 + offering \div base)$;

        end

end
```

Figure 4: Algorithm 3: Shuffling (nd,$\Delta(nd), NDP, \mu, NDP)$)

Its normal form is
{(t_1,3/20),(t_2,3/20),(t_3,3/20),(t_4,3/20,(t_6,2/5)}
Assume nd = {t_1,t_2,t_6,t_9 }, d will be the partial conflict offender since
termset($\hat{d}$)$\cap$nd=[t1,t2,t6]$\neq \emptyset$
{(t_1,3/40),(t_2,3/40),(t_3,13/40),(t_4,13/40,(t_6,1/5)}

The proposed model includes two phases: the training phase and the testing phase. In the training phase, the proposed model first calls PTM (D^+, min sup) to find d-patterns in positive documents (D^{+}) based on a min sup, and evaluates term supports by deploying d-patterns to terms. It also calls Algorithm IP Evolving (D^{-} D^{+}, DP, μ) to revise term supports using noise negative documents in D^{-} based on an experimental coefficient. In the testing phase, it evaluates weights for all incoming documents. The incoming documents then can be sorted based on these weights.

VI. Evaluation And Discussion

6.1 Experimental Data Set

The most popular used data set currently is RCV1, which includes 806,791 news articles for the period between 20 August 1996 and 19 August 1997. These documents were formatted by using a structured XML schema. TREC filtering track has developed and provided two groups of topics (100 in total) for RCV1[10]. The first group includes 50 topics that were composed by human assessors and the second group also includes 50 topics that were constructed artificially from intersections topics. Each topic divided documents into two parts: the training set and the testing set. The training set has a total amount of 5,127 articles and the testing set contains 37,556 articles. Documents in both sets are assigned either positive or negative, where "positive" means the document is relevant to the assigned topic; otherwise "negative" will be shown.

All experimental models use "title" and "text" of XML documents only. The content in "title" is viewed as a paragraph as the one in "text" which consists of paragraphs. For dimensionality reduction, stopword removal is applied and the Porter algorithm [3] is selected for suffix stripping. Terms with term frequency equaling to one are discarded.

6.2 Measures

Several standard measures based on precision and recalls are used. The precision is the fraction of retrieved documents that are relevant to the topic, and the recall is the fraction of relevant documents that have been retrieved.

In addition, the breakeven point (b/p) from graph (IPE AND PTM) comparison is used to provide another measurement for performance evaluation. It indicates the point where the value of precision equals to the value of recall for a topic. The higher the figure of b/p, the more effective the system is. The b/p measure has been frequently used in common information retrieval evaluations.

Evaluation is F_{β} - measure [5], which combines precision and recall and can be defined by the following equation:

$$F\beta - measure = \frac{(\beta 2+1)*precision*recall}{\beta 2*precision+recall}$$

where β is a parameter giving weights of precision and recall and can be viewed as the relative degree of importance attributed to precision and recall [5].

$$F_1 = \frac{2*precision*recall}{precision+recall}$$

6.2.1 Term Frequency Inverse Document Frequency

Term Frequency Inverse Document Frequency (TFIDF) [11] is the most widely adopted measure. TFIDF is the combination of the exhaustively statistic (TF) and the specify statistic (DF) to a term.

TFIDF (t) = TF(d, t) × IDF(t)

6.2.2 Term weighting

Term weighting uses statistical regularities in documents to estimate significance weights for terms. Term weighting functions can measure how specific terms are to a topic by exploiting the statistic variations in the distribution of terms within relevant documents and within a complete document collection [10]. The term weighting strategy should be context-specific [4].

6.2.3 Probabilistic Model

Robertson and I. Soboroff [12] proposed four probabilistic functions for term weighting based on the binary independence retrieval model. Two kinds of assumption are used in these functions: independence assumptions and ordering principles.

6.3 Baseline Models

In order to make a comprehensive evaluation, we choose three classes of models as the baseline models. The first class includes several data mining-based methods.

6.3.1 Concept-Based Models

A new concept-based model was presented in [4] which analyzed terms on both sentence and document levels. This model used a verb-argument structure which split a sentence into verbs and their arguments. Arguments can be further assigned labels such as subjects or objects (or theme). Therefore, a term can be extended and to be either an argument or a verb, and a concept is a labeled term.

For a document d, $tf(c)$ is the number of occurrences of concept c in d; and ctf (c) is called the conceptual term frequency of concept c in a sentence s, which is the number of occurrences of concept c in the verb-argument structure of sentence s. Given a concept c, its tf and ctf can be normalized as $tf_{weight}(c)$ and $ctf_{weight}(c)$, and its weight can be evaluated as follows

$$Tf_{weight}(c) = Tf_{weight}(c) + c\,Tf_{weight}(c)$$

To have a uniform representation, we call a concept as a concept-pattern which is a set of terms. For example, verb “hits” is denoted as {hits} and its argument “the ball” is denoted as {hits}

“the ball” with the concept-based model , we design a concept based model (CBM) for describing the features in a set of positive documents, which consists of two step.The first step is to find all of the concepts in the positive documents of the training set, where verbs are extracted from PropBank data set at http://verbs.colorado.edu/verb-index/propbank-1.0.tar. gz. The second step is to use the deploying approach to evaluate the weights of terms based on their appearances in these discovery concepts. Since all positive documents are treated equally before the process of document evaluation, the value of α_i is set as 1.0 for all of the positive documents and thus the α_i value for the negative documents can be determined by using (5).

In document evaluation, once the concept for a topic is obtained, the similarity between a test document and the concept is estimated using inner product. The relevance of a document d to a topic can be calculated by the function R (d) = d.c where d is the term of d and c is the concept of the topic.

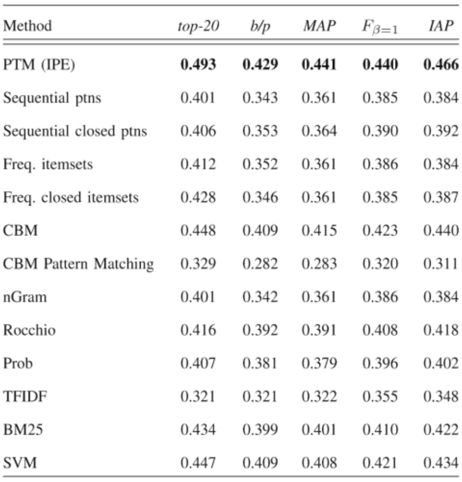
6.4 Hypotheses

The major objective of the experiments is to show how the proposed approach can help improving the effectiveness of pattern-based approaches. Experiments involve comparing the performance of different pattern-based models, concept-based models, and term- based models.

Hypothesis H1: The proposed model, PTM (IPE), is designed to achieve the high performance for determining relevant information to answer what users want. The model would be better than other pattern- based models, concept-based models, and state-of- the-art term-based models in the effectiveness.

Hypothesis H2: The proposed deploying method has better performance for the interpretation of discovered patterns in text documents. This deploying approach is not only promising for pattern-based approaches, but also significant for the concept- based model.

categories defined : The first category contains all data mining-based (DM) methods, such as sequential pattern mining, sequential closed pattern mining, frequent itemset mining, frequent closed itemset mining, where min sup = 0.2. The second category includes the concept-based model that uses the deploying method and the CBM Pattern Matching model; and the last category includes nGram, Rocchio, Probabilistic model, TFIDF, and two state-of-the- art models, BM25 and SVM. A brief of these methods is depicted in Table 4.



Method	*top-20*	*b/p*	*MAP*	$F_{\beta=1}$	*IAP*
PTM (IPE)	**0.493**	**0.429**	**0.441**	**0.440**	**0.466**
Sequential ptns	0.401	0.343	0.361	0.385	0.384
Sequential closed ptns	0.406	0.353	0.364	0.390	0.392
Freq. itemsets	0.412	0.352	0.361	0.386	0.384
Freq. closed itemsets	0.428	0.346	0.361	0.385	0.387
CBM	0.448	0.409	0.415	0.423	0.440
CBM Pattern Matching	0.329	0.282	0.283	0.320	0.311
nGram	0.401	0.342	0.361	0.386	0.384
Rocchio	0.416	0.392	0.391	0.408	0.418
Prob	0.407	0.381	0.379	0.396	0.402
TFIDF	0.321	0.321	0.322	0.355	0.348
BM25	0.434	0.399	0.401	0.410	0.422
SVM	0.447	0.409	0.408	0.421	0.434

Table 4: Comparison of All Methods on the First 50 Topics

6.5 Experimental Results

This section presents the results for the evaluation of the proposed approach PTM (IPE), inner pattern evolving in the pattern taxonomy model. . In addition, results obtained based on the first 50 TREC topics are more practical and reliable since the judgment for these topics is manually made by domain experts, whereas the judgment for the last 50 TREC topics is created based on the metadata tagged in each document is viewed in fig. 5 The most important information revealed in this table 6 is that our proposed PTM (IPE) outperforms not only the pattern mining-based methods, but also the term-based methods including the state-of-the-art methods BM25 and SVM. PTM (IPE) also out performs. For the time complexity in the testing phase, all models take $O (|T| \times |d|)$for all incoming documents d. In our experiments, all models used 702 terms for each topic in average. Therefore, there is no significant difference between these models on time complexity in the testing phase.

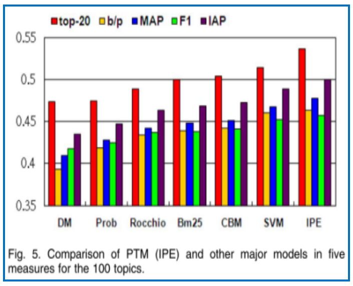


Fig. 5. Comparison of PTM (IPE) and other major models in five measures for the 100 topics.

Figure:5: Comparison of PTM(IPE) and other major models in five measures for the 100 topics

	PDM	IPE$_{\mu=5}$
top-20	0.5265	**0.5360**
b/p	0.4598	**0.4632**
MAP	0.4734	**0.4770**
$F_{\beta=1}$	0.4528	**0.4570**
IAP	0.4932	**0.4994**

Table 5: Performance of Inner Pattern evolving in PTM on all Topics

The formula results us the detail view of user required document by avoiding the noisy patterns at certain comfort level.

$$\text{Ratio} = \frac{|\{d|d \in D-, weight(d) \geq threshold(DP)\}|}{|D+|+|D\_|}$$

Where : D $^{+}$ describes Positive documents
D – describes Negative documents
DP – discovered Patterns

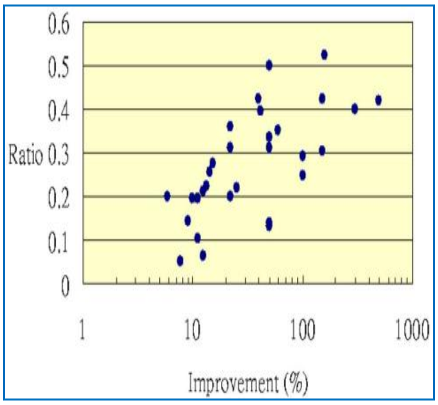


Figure 6: The relationship between the proportion in number of negative documents greater than threshold to all documents and corresponding improvement on IPE with µ=5 on improved topics

VII. Conclusions

Many data mining techniques include association rule mining, frequent item-set mining, sequential pattern mining, maximum pattern mining, and closed pattern mining in the past. The reason is that some useful long patterns with high specificity lack in support (i.e., the low-frequency, misinterpretation problem). Hence, mis interpretations of patterns derived from data mining techniques lead to the ineffective performance. In this paper, an effective pattern discovery technique has been proposed to overcome the low frequency and misinterpretation problems of text mining.

References

[1] K. Aas and L. Eikvil, "Text Categorisation: A Survey," Technical Report Raport NR 941, Norwegian Computing Center, 1999.

[2] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," Proc. 20th Int'l Conf. Very Large Data Bases (VLDB '94), pp. 478-499, 1994.

[3] H.Ahonen, O.Heinonen, M.Klemettinen, and A.I. Verkamo, "Applying Data Mining Techniques for Descriptive Phrase Extraction in Digital Document Collections,"Proc.IEEE Int'l Forum on Research and Technology Advances in Digital Libraries (ADL '98), pp. 2-11, 1998.

[4] M.F.Caropreso, S. Matwin, and F. Sebastiani, "Statistical Phrases in Automated Text Categorization," Technical Report IEI-B4-07- 2000, Instituto di Elaborazione dell'Informazione, 2000.

[5] S.T. Dumais, "Improving the Retrieval of Information from External Sources," Behavior

Research Methods, Instruments, and Computers, vol. 23, no. 2, pp. 229-236, 1991.

[6] D.D. Lewis, "An Evaluation of Phrasal and Clustered Representa- tions on a Text Categorization Task," Proc. 15th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '92), pp. 37-50, 1992.

[7] X. Li and B. Liu, "Learning to Classify Texts Using Positive and Unlabeled Data," Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI'03), pp. 587-594, 2003.

[8] Y. Li and N. Zhong, "Interpretations of Association Rules by Granular Computing," Proc. IEEE Third Int'l Conf. Data Mining (ICDM '03), pp.593-596, 2003.

[9] M.F. Porter, "An Algorithm for Suffix Stripping," Program, vol. 14, no. 3, pp. 130-137, 1980.

[10] H. Lodhi, C.Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text Classification Using String Kernels," J. Machine Learning Research, vol. 2, pp. 419-444, 2002.

[11] Sheng "Knowledge Discovered Using PTM in Text Mining " .

[12] M.F.Porter ,"An algorithm for suffix Striping," Program ,vol 14,no.3,pp.130-137,1980.