International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org Volume 2 Issue 1 ǁ January. 2013 ǁ PP.01-10

Mining The Text Documents Using Phrase Based Tokenizer Approach

[1]V.Umadevi, [2]Dr.Mrs.D.shanthi,

[1,2]M.E.Computer Science & Engineering, PSNA College of Engineering and Technology, Dindigul

ABSTRACT: *Text mining, sometimes alternately referred to as text data mining, roughly equivalent to text analytics, refers to the process of deriving high-quality information from text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning. Text mining is the discovery of interesting knowledge in text documents. Many data mining techniques have been proposed for mining useful patterns in text documents. It is a challenging issue to find accurate knowledge (or features) in text documents to help users to find what they want. In existing, Information Retrieval (IR) provided many term-based methods to solve this challenge. The term-based methods suffer from the problems of polysemy and synonymy. The polysemy means a word has multiple meanings, and synonymy is multiple words having the same meaning. In proposed to use pattern (or phrase)-based approaches should perform better than the term-based ones. The proposed approach can improve the accuracy of evaluating term weights because discovered patterns are more specific than whole documents.*

Index Terms— *Text mining, text classification, pattern mining, pattern evolving, information filtering.*

I. INTRODUCTION

The knowledge discovery and data mining have attracted a great deal of attention with an imminent need for turning such data into useful information and knowledge. Many application, such as market analysis and business management, can benefit by the use of the information and knowledge extracted from a large amount of data. Knowledge discovery can be viewed as the process of nontrivial extraction of information from large databases, information that is implicitly presented in the data, previously unknown and potentially useful for users. Data mining is therefore an essential step in the process of knowledge discovery in databases.

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. Most of them are proposed for the purpose of developing efficient mining algorithms to find particular patterns within a reasonable and acceptable time frame. With a large number of patterns generated by using data mining approaches, how to effectively use and update these patterns is still an open research issue. In this paper, we focus on the development of a knowledge discovery model to effectively use and update the discovered patterns and apply it to the field of text mining.

Information Retrieval (IR) provided many term-based methods to solve this challenge, such as Rocchio and probabilistic models, rough set models, BM25 and support vector machine (SVM) based filtering models. The advantages of term based methods include efficient computational performance as well as mature theories for term weighting, which have emerged over the last couple of decades from the IR and machine learning communities. However, term based methods suffer from the problems of polysemy and synonymy, where polysemy means a word has multiple meanings, and synonymy is multiple words having the same meaning. The phrase-based approaches could perform better than the term based ones, as phrases may carry more "semantics" like information. This hypothesis has not fared too well in the history of IR. Although phrases are less ambiguous and more discriminative than individual terms, the likely reasons for the discouraging performance include: 1) phrases have inferior statistical properties to terms, 2) they have low frequency of occurrence, and 3) there are large numbers of redundant and noisy phrases among them.

There are two fundamental issues regarding the effectiveness of pattern-based approaches: low frequency and misinterpretation. Given a specified topic, a highly frequent pattern (normally a short pattern with large support) is usually a general pattern, or a specific pattern of low frequency .If we decrease the minimum support, a lot of noisy patterns would be discovered. Misinterpretation means the measures used in pattern mining (e.g., "support" and "confidence") turn out to be not suitable in using discovered patterns to answer what users want. The difficult problem hence is how to use discovered patterns to accurately evaluate the weights of useful features (knowledge) in text documents.

To overcome the disadvantages of phrase-based approaches, pattern mining-based approaches (or pattern taxonomy models (PTM) have been proposed, which adopted the concept of closed sequential patterns, and pruned non closed patterns. These pattern mining-based approache have shown certain extent improvements on the effectiveness. However, the paradox is that people think pattern-based approaches could be a significant alternative, but consequently less significant improvements are made for the effectiveness compared with term-based methods. To solve the above paradox, this paper present Effective Pattern discovery technique, which first

Calculates discovered specificities of patterns and then evaluates term weights according to the distribution of terms in the discovered patterns rather than the distribution in documents for solving the misinterpretation problem. It also considers the influence of patterns from the negative training examples to find ambiguous (noisy) patterns and try to reduce their influence for the low-frequency problem. The process of updating ambiguous patterns can be referred as pattern evolution. The proposed approach can improve the accuracy of evaluating term weights because discovered patterns are more specific than whole documents.

II. RELATED WORK

Many types of text representations have been proposed in the past. A well known one is the bag of words that uses keywords (terms) as elements in the vector of the feature space. In the tf*idf weighting scheme is used for text representation in Rocchio classifiers. In addition to TFIDF, the global IDF and entropy weighting scheme is proposed in and improves performance by an average of 30 percent. Various weighting schemes for the bag of words representation approach were given in .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 over fitting .In order 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 choice of a representation depended on what one regards as the meaningful units of text and the meaningful natural language rules for the combination of these units. With respect to the representation of the content of documents ,some research works have used phrases rather than individual words. In the combination of unigram and bigrams was chosen for document indexing in text categorization (TC) and evaluated on a variety of feature evaluation functions (FEF). A phrase-based text representation for Web document management was also proposed in.

Data mining techniques have been used for text analysis by extracting co occurring terms as descriptive phrases from document collections. However, the effectiveness of the text mining systems using phrases as text representation showed no significant improvement. The likely reason was that a phrase-based method had "lower consistency of assignment and lower document frequency for terms" as mentioned in Term-based ontology mining methods also provided some thoughts for text representations. For example, hierarchical clustering was used to determine synonymy and hyponymy relations between keywords .Also, the pattern evolution technique was introduced. In order to improve the performance of term-based ontology mining.

Pattern mining has been extensively studied in data mining communities for many years. A variety of efficient algorithms such as Apriori-like algorithms, PrefixSpan, FP-tree, SPADE, SLPMiner, and GST have been proposed. These research works have mainly focused on developing efficient mining algorithms for discovering patterns from a large data collection. However, searching for useful and interesting patterns and rules was still an open problem. In the field of text mining, pattern mining techniques can be used to find various text patterns, such as sequential patterns, frequent item sets, co-occurring terms and multiple grams, for building up a representation with these new types of features. The challenging issue is how to effectively deal with the large amount of discovered patterns.

For the challenging issue, closed sequential patterns have been used for text mining in , which proposed that the concept of closed patterns in text mining was useful and had the potential for improving the performance of text mining. Pattern taxonomy model was also developed in and to improve the effectiveness by effectively using closed patterns in text mining. In addition, a two-stage model that used both term-based methods and pattern based methods was introduced in to significantly improve the performance of information filtering.

Natural language processing (NLP) is a modern computational technology that can help people to understand the meaning of text documents. For a long time, NLP was struggling for dealing with uncertainties in human languages. Recently, a new concept-based model was presented to bridge the gap between NLP and text mining, which analyzed terms on the sentence and document levels .This model included three components. The first component analyzed the semantic structure of sentences; the second component constructed a conceptual ontological graph (COG) to describe the semantic structures; and the last component extracted top concepts based on the first two components to build feature vectors using the standard vector space model. The advantage of the concept-based model is that it can effectively discriminate between non important terms and meaningful

terms which describe a sentence meaning. Compared with the above methods, the concept-based model usually relies upon its employed NLP techniques.

III. PATTERN TAXONOMY MODEL

Assume that all documents are split into paragraphs. So a given document d yields a set of paragraphs PS(d). Let D be a training set of documents, which consists of a set of positive documents, D+; and a set of negative documents, D_. Let T ={t1, t2, . . . ,tm}be a set of terms (or keywords) which can be extracted from the set of positive documents, D+.

3.1. Frequent and Closed Patterns

Given a term set X in document d,-X is used to denote the covering set of X for d, which includes all paragraphs dp ∈PS(d) such that X ⊆ dp, i.e., X={dp | dp ∈ PS(d),X ⊆ dp). Its absolute support is the number of occurrences of X in PS(d), that is supa (X) = |X| . Its relative support is the fraction of the paragraphs that contain the pattern, that is, supr (X) =|X|/PS(d)| .

A term set X is called frequent pattern if its supr (or supa)>= min _sup, a minimum support. Table 1 lists a set of paragraphs for a given document d, where PS(d) ={dp1, dp2, . . . ,dp6}, and duplicate terms were removed .Let min_sup = 50%,we can obtain ten frequent patterns in Table 1 using above definitions. Table 2 illustrates the ten frequent patterns and heir covering sets.

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

Not all frequent patterns in Table 2 are useful. For example, pattern {t3, t4} always occurs with term t6 in paragraphs, i.e., the shorter pattern, {t3; t4}, is always a part of the larger pattern, {t3; t4; t6}, in all of the paragraphs. Hence, we believe that the shorter one, {t3; t4}, is a noise pattern and expect to keep the larger pattern, {t3; t4; t6}, only.

Given a term set X, its covering set X is a subset of paragraphs. Similarly, given a set of paragraphs Y ⊆ PS(d),
we can define its term set, which satisfies

$$termset(Y) = \{t | \forall dp \in Y => t \in dp\}.$$

The closure of X is defined as follows:

$$Cls(X) = termset(\ulcorner X \urcorner).$$

A pattern X (also a term set) is called closed if and only if X = Cls(X).

Let x be a closed pattern .we can prove that

$$sup_a(X_1) < sup_a(X), \quad (1)$$

For all patterns X1 $\supset$ X otherwise ,if $sup_a(X_1) = sup_a(X)$,

We have

$$\ulcorner X_1 \urcorner = \ulcorner X \urcorner,$$

Where $sup_a(X_1)$ and $sup_a(X)$ are the absolute support of pattern X1 and x, respectively.

We also have

$$Cls(X) = termset(\ulcorner X \urcorner) = termset(\ulcorner X_1 \urcorner) \supseteq X_1 \supset X,$$ that is, $Cls(X) \neq X.$

3.2. Pattern Taxonomy

Patterns can be structured into a taxonomy by using the is-a (or subset) relation. For the example of Table 1, where we have illustrated a set of paragraphs of a document, and the discovered 10 frequent patterns in Table 2 if assuming min sup ¼ 50%. There are, however, only three

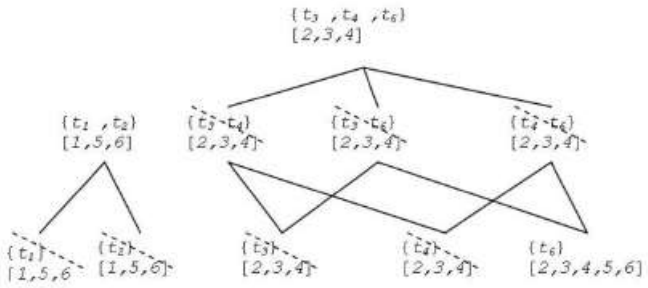

Fig 2 Pattern taxonomy

Closed patterns in this example. They are <t3, t4, t6>, <t1, t2>, and <t6>.

Fig. 2 illustrates an example of the pattern taxonomy for the frequent patterns in Table 2, where the nodes represent frequent patterns and their covering sets; non closed patterns can be pruned; the edges are "is-a" relation. After pruning, some direct "is-a" retaliations may be changed, for example, pattern ft6g would become a direct sub pattern of {t3, t4, t6} after pruning non closed patterns.

Smaller patterns in the taxonomy, for example pattern {t6}, (see Fig. 2) are usually more general because they could be used frequently in both positive and negative documents; and larger patterns, for example pattern {t3, t4, t6}, in the taxonomy 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,.

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 (see the definition below) in order to accurately evaluate term weights (supports).

The rational behind this motivation is that d-patterns include more semantic meaning than terms that are selected based on a term-based technique (e.g., tf*idf). As a result, a term with a higher tf*idf value could be meaningless if it has not cited by some d-patterns (some important parts in documents). The evaluation of term weights (supports) is different to the normal term-based approaches. In the term-based approaches, the evaluation of term weights are based on the distribution of terms in documents. In this research, terms are weighted according to their appearances in discovered closed patterns.

4.1 Representation of closed patterns

It is complicated to derive a method to apply discovered patterns in text documents for information filtering systems. To simplify this process, we first review the composition operation $\oplus$ defined.

Let p1 and p2 be sets of term-number pairs. p1 $\oplus$ p2 is called the composition of p1 and p2 which satisfies

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

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.

For example,

$$\{(t_1, 1), (t_2, 2), (t_3, 3)\} \oplus \{(t_2, 4)\} = \{(t_1, 1), (t_2, 6), (t_3, 3)\},$$

or

$$\{(t_1, 2\%), (t_2, 5\%), (t_3, 9\%)\} \oplus \{(t_1, 1\%), (t_2, 3\%)\} = \{(t_1, 3\%), (t_2, 8\%), (t_3, 9\%)\}.$$

Formally, for all positive documents di $\in$ D+, we first deploy its closed patterns on a common set of terms T in order to obtain the following d-patterns (deployed patterns, nons equential weighted patterns):

$$\widehat{d}_i = \{(t_{i_1}, n_{i_1}), (t_{i_2}, n_{i_2}), \ldots, (t_{i_m}, n_{i_m})\}, \quad (2)$$

where tij in pair (tij, nij) denotes a single term and nij is its support in di which is the total absolute supports given by closed patterns that contain tij; or nij (simply in this paper) is the total number of closed patterns that contain tij .

For example, using Fig. 1 and Table 1, we have

$$sup_a(<t_3, t_4, t_6>) = 3,$$
$$sup_a(<t_1, t_2>) = 3,$$
$$sup_a(<t_6>) = 5, \text{ and}$$
$$\widehat{d} = \{(t_1, 3), (t_2, 3), (t_3, 3), (t_4, 3), (t_6, 8)\}.$$

The process of calculating d-patterns can be easily described by using the $\oplus$ operation in Algorithm 1 (PTM) shown in Fig. 3 that will be described in the next section, where a term's support is the total number of closed patterns that contain the term.

Table 3 illustrates a real example of pattern taxonomy for a set of positive documents.

We also can obtain the d-patterns of the five sample documents in Table 3 which are expressed as follows:

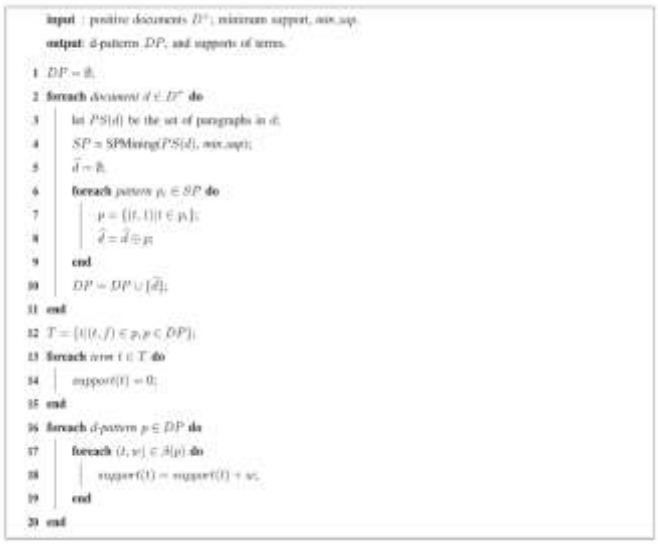

Fig 3. Algorithm 1: PTM (D+,min_sup).

$$\widehat{d_1} = \{(carbon, 2), (emiss, 1), (air, 1), (pollut, 1)\},$$
$$\widehat{d_2} = \{(greenhous, 1), (global, 2), (emiss, 1)\},$$
$$\widehat{d_3} = \{(greenhous, 1), (global, 1), (emiss, 1)\},$$
$$\widehat{d_4} = \{(carbon, 1), (air, 2), (antarct, 1)\},$$
$$\widehat{d_5} = \{(emiss, 1), (global, 1), (pollut, 1)\}.$$

4.2 D-pattern Mining Algorithm

To improve the efficiency of the pattern taxonomy mining, an algorithm, SPMining, to find all closed sequential patterns, which used the well-known Apriori property in order to reduce the searching space.Algorithm 1 (PTM) shown in Fig. 3 describes the training process of finding the set of d-patterns. For every positive document, the SPMining algorithm is first called in step 3 giving rise to a set of closed sequential patterns SP. The main focus of this paper is the deploying process, which consists of the d-pattern discovery and term support evaluation. In Algorithm 1 (Fig. 3), all discovered patterns in a positive document are composed into a d_pattern giving rise to a set of d-patterns DP in steps 5 to 8. Thereafter, from steps 11 to 18, term supports are calculated based on the normal forms for all terms in d_patterns.

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 5 to 8) is O(Kk2n). Step 9 takes O(mn). Step 11 also gets all terms from d-patterns and takes O(m2n2). Steps 12 to 14 initialize support function and take O (m), and the steps 15 to 19 take O(mn). Therefore, the time complexity of pattern deploying is

TABLE 3
Example of a Set of Positive Documents Consisting of Pattern Taxonomies

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	$PT_{(5,1)}$	$\{\langle emiss, global, pollut\rangle_2\}$

The number beside each sequential pattern indicates the absolute support of pattern.

$$O(Kk^2n + mn + m^2n^2 + m + mn) = O(Kk^2n + m^2n^2).$$

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

$$weight(d) = \sum_{t \in T} support(t)\tau(t, d), \quad (3)$$

where support(t) is defined in Algorithm (Fig. 3); and T(t,d) = 1 if t $\in$ d; otherwise T(t,d) = 0.

V. INNER PATTERN EVOLUTION

In this section, described 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:

$$Threshold(DP) = \min_{p \in DP} \left(\sum_{(t,w) \in \beta(p)} support(t) \right). \quad (4)$$

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.

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:

$$\Delta(nd) = \{p \in DP | termset(p) \cap nd \neq \emptyset\}.$$

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 basic idea of updating patterns is explained as follows: complete conflict offenders are removed from d-patterns first. For partial conflict offenders, their term supports are reshuffled in order to reduce the effects of noise documents.

The main process of inner pattern evolution is implemented by the algorithm IP Evolving (see Algorithm 2 in Fig. 4). The inputs of this algorithm are a set of d-patterns DP, a training set $D = D^+ \cup D^-$. The output is a composed of d-pattern. Step 2 in IP Evolving is used to estimate the threshold for finding the noise negative documents. Steps 3 to 10 revise term supports by using all noise negative documents. Step 4 is to find noise documents and the corresponding offenders. Step 5 gets normal forms of dpatterns NDP. Step 6 calls algorithm Shuffling (see Algorithm 3 in Fig. 5) to update NDP according to noise documents. Steps 7 to 9 compose updated normal forms together.

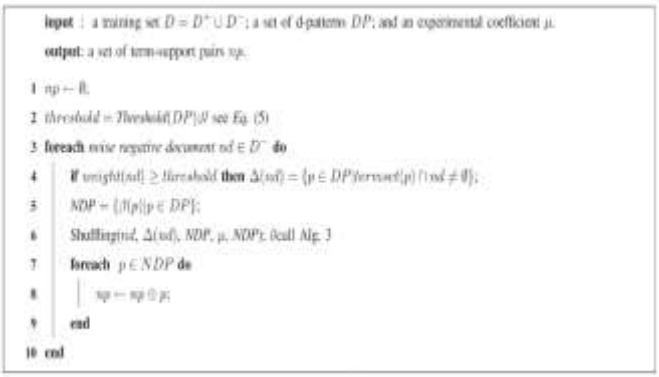

Fig 4 Algorithm 2: IP Evolving

The time complexity of Algorithm 2 in Fig. 4 is decided by step 2, the number of calls for Shuffling algorithm and the number of using $\oplus$ operation. Step 2 takes O(nm). For each noise negative pattern nd, the algorithm gets its offenders that takes $O(nm \times |nd|)$ in step 4, and then calls once Shuffling. After that, it calls n $\oplus$ operation that takes $O(nmm) = O(nm^2)$.

The parameter offering is used in step 4 for the purpose of temporarily storing the reduced supports of some terms in a partial conflict offender. The offering is part of the sum of supports of terms in a d-pattern where these terms also appear in a noise document. The algorithm calculates the base in step 5 which is certainly not zero since $termset(p) - nd \neq \emptyset$; and then updates the support distributions of terms in step 6.

For example, for the following d-pattern

$$\widehat{d} = \{(t_1, 3), (t_2, 3), (t_3, 3), (t_4, 3), (t_6, 8)\}.$$

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 = {t1,t2,t6,t9}, $\widehat{d}$ will be a partial conflict offender since

$$termset(\widehat{d}) \cap nd = \{t_1, t_2, t_6\} \neq \emptyset.$$

Let $\mu = 2$, offering=1/2*(3/20+3/20+2/5) = 7/20, and base = 3/20+3/20=3/10.Hence, we can get the following updated normal form by using algorithm shuffling:

$$\{(t_1, 3/40), (t_2, 3/40), (\mathbf{t_3}, 13/40), (\mathbf{t_4}, 13/40), (t_6, 1/5)\}.$$

Let m =|T|, n =|D+|the number of positive documents in a training set, and q be the number of noise negative documents in D-. The time complexity of algorithm Shuffling is decided by steps 6 to 9. For a given noise negative document nd, its time complexity is O(nm2) if let $nd = nd \cap T$, where $T = \{t \in termset(p) | p \in DP\}$. Hence, the time complexity of algorithm Shuffling is O(nm2) for a given noise negative document.

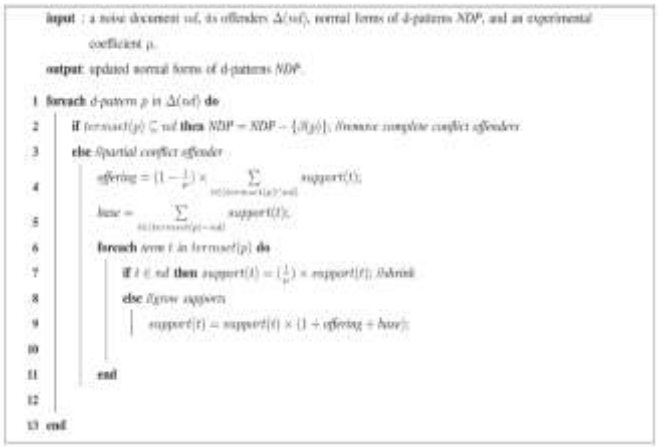

Fig 5 Algorithm 3: Shuffling

Based on the above analysis about Algorithms 2 and 3, the total time complexity of the inner pattern evolution is $O(nm + q(nm|nd| + nm^2) + nm^2) = O(qnm^2)$ considering that the noise negative document nd can be replaced by nd $\cap$ T before conducting the pattern evolution.

The proposed model includes two phases: the training phase and the testing phase. In the training phase, the proposed model first calls Algorithm PTM (D+, min _sup) to find d-patterns in positive documents (D+) based on a min _sup, and evaluates term supports by deploying dpatterns 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 μ. In the testing phase, it evaluates weights for all incoming documents using eq. (4). The incoming documents then can be sorted based on these weights.

VI. PERFORMANCE EVALUATI ON

In this section represent, Reuters text collection is used to evaluate the proposed approach. Term stemming and stop word removal techniques are used in the prior stage of text preprocessing. Several common measures are then applied for Performance evalution and our results are compared with the state-of-art approaches in data mining, concept-based, and term-based methods.

6.1 Concept based method

The concept-based method was presented in 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. For example, "John hits the ball," where "hits" is a verb, and "John" or "the ball" are the arguments of "hits." 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 ct f(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 tfweight(c) and ctfweight(c), and its weight can be evaluated as follows:

$$weight(c) = tf_{weight}(c) + ctf_{weight}(c).$$

To have a uniform representation, in this paper, we call a concept as a concept-pattern which is a set of terms. For example, verb "hits" is denoted as fhitsg and its argument "the ball" is denoted as {the; ball}

6.2 Term based method

There are many classic term-based approaches. The Rocchio algorithm which has been widely adopted in information retrieval, can build text representation of a training set using a Centroid ~c as follows:

$$\vec{c} = \alpha \frac{1}{|D^+|} \sum_{\vec{d} \in D^+} \frac{\vec{d}}{\|\vec{d}\|} - \beta \frac{1}{|D^-|} \sum_{\vec{d} \in D^-} \frac{\vec{d}}{\|\vec{d}\|},$$

where α and β are empirical parameters; D+ and D- are the sets of positive and negative documents, respectively; $\vec{d}$ denotes a document.

For both term-based models and CBM, we use the following equation to assign weights for all incoming documents d based on their corresponding W functions

$$weight(d) = \sum_{t \in T} W(t)\tau(t, d).$$

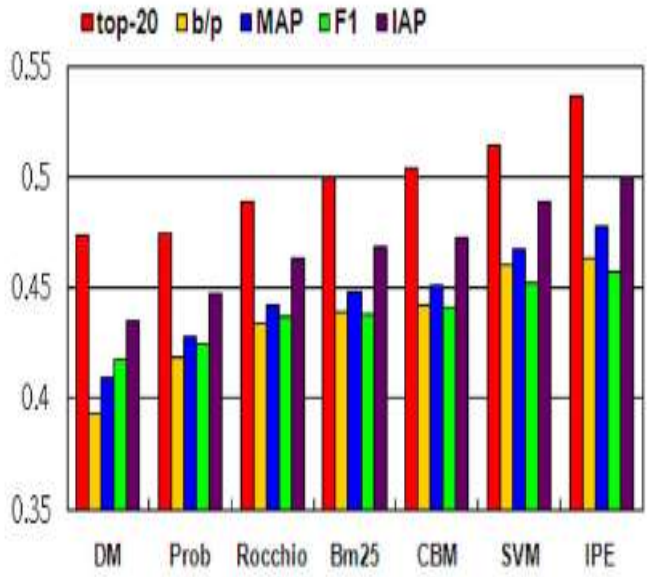

Fig 6.Comparision of PTM (IPE) and other major models in five measures for the 100 topics.

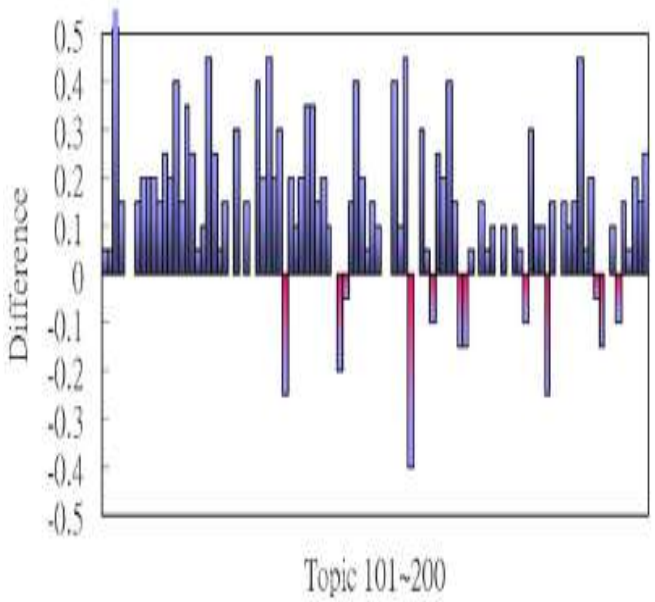

Fig 7.comparision of ptm (ipe) and tfidf in top-20 precision.

VII. CONCLUSION

Many data mining techniques have been proposed in the last decade. These techniques include association rule mining, frequent item set mining, sequential pattern mining, maximum pattern mining, and closed pattern mining. However, using these discovered knowledge (or patterns) in the field of text mining is difficult and ineffective. The reason is that some useful long patterns with high specificity lack in support (i.e., the low-frequency problem). We argue that not all frequent short patterns are useful. Hence, misinterpretations of patterns derived from data mining techniques lead to the ineffective performance. In this research work, an effective pattern discovery technique has been proposed to overcome the low-frequency and misinterpretation problems for text mining. The proposed technique uses two processes, pattern deploying and pattern evolving, to refine the discovered patterns in text documents. The experimental results show that the proposed model outperforms not only other pure data mining-based methods and the concept based model, but also term-based state-of-the-art models, such as BM25 and SVM-based models.

REFERENCES

[1]. [1] Y.Li, C. Zhang, and J.R. Swan, "An Information Filtering Modelon the Web and Its Application in Jobagent," Knowledge-Based Systems, vol. 13, no. 5, pp. 285-296, 2000.

[2]. [2] 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.

[3]. [3] 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.

[4]. [4] Y. Li, W. Yang, and Y. Xu, "Multi-Tier Granule Mining for Representations of Multidimensional Association Rules," Proc. IEEE Sixth Int'l Conf. Data Mining (ICDM '06), pp. 953-958, 2006.

[5]. [5] Y. Li and N. Zhong, "Mining Ontology for Automatically

[6]. Acquiring Web User Information Needs," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 4, pp. 554-568, Apr. 2006.

[7]. [6] Y. Li, X. Zhou, P. Bruza, Y. Xu, and R.Y. Lau, "A Two-Stage Text Mining Model for Information Filtering," Proc. ACM 17th Conf. Information and Knowledge Management (CIKM '08), pp. 1023-1032,2008.

[8]. [7] 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.

[9]. [8] A. Maedche, Ontology Learning for the Semantic Web. Kluwer Academic, 2003.

[10]. [9] C. Manning and H. Schu¨ tze, Foundations of Statistical Natural Language Processing. MIT Press, 1999.

[11]. [10] I. Moulinier, G. Raskinis, and J. Ganascia, "Text Categorization: A Symbolic Approach," Proc. Fifth Ann. Symp. Document Analysis and Information Retrieval (SDAIR), pp. 87-99, 1996.

[12]. [11] J.S. Park, M.S. Chen, and P.S. Yu, "An Effective Hash-Based Algorithm for Mining Association Rules," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '95), pp. 175-186, 1995.

[13]. [12] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "Prefixspan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth," Proc. 17th Int'l Conf. Data Eng. (ICDE '01), pp. 215-224, 2007.

[14]. [13] M.F. Porter, "An Algorithm for Suffix Stripping," Program, vol. 14, no. 3, pp. 130-137, 2009.

[15]. [14] S.E. Robertson, S. Walker, and M. Hancock-Beaulieu, "Experimentation as a Way of Life: Okapi at Trec," Information Processing and Management, vol. 36, no. 1, pp. 95-108, 2000.

[16]. [15] T. Rose, M. Stevenson, and M. Whitehead, "The Reuters Corpus Volume1—From Yesterday's News to Today's Language Resources," Proc. Third Int'l Conf. Language Resources and Evaluation, pp. 29-31, 2002.

[17]. [16] F. Sebastiani, "Machine Learning in Automated Text Categorization,"

[18]. ACM Computing Surveys, vol. 34, no. 1, pp. 1-47, 2002.

[19]. [17] M. Seno and G. Karypis, "Slpminer: An Algorithm for Finding Frequent Sequential Patterns Using Length-Decreasing Support Constraint," Proc. IEEE Second Int'l Conf. Data Mining (ICDM '02), pp. 418-425, 2002.