VOL. 13, NO. 11, JUNE 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

EVALUATION AND ANALYSIS OF DISCOVERED PATTERNS USING PATTERN CLASSIFICATION METHODS IN TEXT MINING

Ravindra Changala and D. Rajeswara Rao
Department of Computer Science Engineering, K L University, Andhra Pradesh, India
E-Mail: rtit2009@gmail.com

ABSTRACT

Pattern Deploying Methods performed good in discovering knowledge. These methods have given accurate results. Still it is observed that few of discovered patterns are holding noise knowledge instead of required and low frequency problem of long patterns. Hence we focused on perfect evaluation of discovered patterns by adapting the concepts of Deployed Pattern Evaluation (DPE) and Individual Pattern Evaluation (IPE). We used closed sequential algorithms to use the semantic information in the patterns to improve the performance and for accurate term weights we used d-patterns which use the evaluations of term weights based on the distribution of terms in documents. In this paper, terms are weighted according to their appearances in discovered closed patterns. Pattern Classification Models (PCM) Pattern Deploying Methods (PDS) resolved some extent the problems with low-frequency patterns. But still there is gap of pattern usage effectively can be resolved by our new approach. We also concentrated on ambiguous patterns influences in the documents. We made an analysis in comparison of other algorithms and methods hence our approach proved better.

Keywords: PTM, IPE, PDS, D-patterns, closed sequential pattern, text mining, CBM.

1. INTRODUCTION

We are rich in data, poor in information; the solution for this problem is data mining by extending data mining techniques. 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.

Text mining is the discovery of interesting knowledge 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. Information Retrieval (IR) served some extent with term-based methods and also Rocchio and probabilistic models [4], rough set models [23], BM25 and support vector machine (SVM) based filtering models.

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.

We also conduct numerous experiments on the latest data collection, Reuters Corpus Volume 1 (RCV1) and Text Retrieval Conference (TREC) filtering topics, to evaluate the proposed technique. The results show that the proposed technique outperforms up-to-date data mining-based methods, concept-based models and the state-of-the-art term based methods. We also conduct numerous experiments on the latest data collection, Reuters Corpus Volume 1 (RCV1) and Text Retrieval Conference (TREC) filtering topics, to evaluate the proposed technique. The results show that the proposed technique outperforms up-to-date data mining-based methods, concept-based models and the state-of-the-art term based methods.

2. 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 [21], 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 [9] and improves performance by an average of 30 percent. Various weighting schemes for the bag of words representation approach were given in [1], [14].

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. Details of these selection functions were stated in [19].

In [3], 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 [18].

Pattern mining has been extensively studied in data mining communities for many years. A variety of efficient algorithms such as Apriori-like algorithms [2], PrefixSpan, FP-tree [10], [11], SPADE, SLPMiner, and GST [12] 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 [22], [24]. 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. Nevertheless, the challenging issue is how to effectively deal with the large amount of discovered patterns.

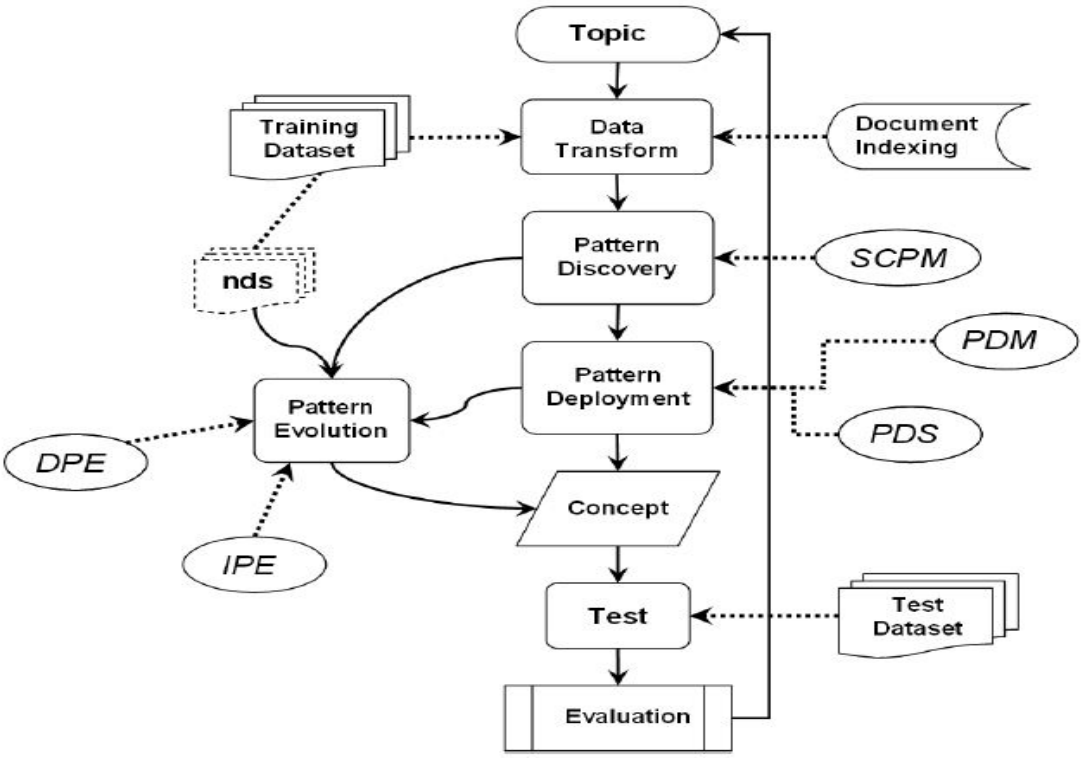

Figure-1. Procedure for pattern deploying methods [1].

For the challenging issue, closed sequential patterns have been used for text mining in [15], 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 [14] and [19] 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 [16] to significantly improve the performance of information filtering.

3. PROBLEM FORMULATION

Instead of the keyword-based concept used in the traditional document representation model, the pattern based model containing frequent sequential patterns (single term or multiple terms) is used to perform the same concept of task. This section will define the basic problem of mining sequential pattern in text documents

Basic definition

The basic definition of sequences used in this study is described as follows. Let $T = \{t1, t2, \ldots, tk\}$ be a set of all terms, which can be viewed as keywords in text datasets. A sequence $S = \langle s1, s2, \ldots, sn \rangle$ ($si \in T$) is an ordered list of terms. A sequence $\alpha = \langle a1, a2, \ldots, an \rangle$ is a sub-sequence of another sequence $\beta = \langle b1, b2, \ldots, bm \rangle$, denoted by $\alpha \subseteq \beta$, if there exist integers $1 \leq i1 < i2 < \ldots < in \leq m$, such that $a1 = bi1$, $a2 = bi2, \ldots, an = bin$. The sequence α is a *proper* sub-sequence of β if $\alpha \subseteq \beta$ but $\alpha \neq \beta$, denoted by $\alpha \subset \beta$. For instance, sequence <A, C> is a sub-sequence of sequences <A, B, C>. However, <B, A> is not a sub-sequence of <A, B, C> since the order of terms is considered. In addition, we also can say sequence <A, B, C> is a super-sequence of <A, C>. The problem of mining sequential patterns is to find the complete set of sub-sequences from a set of sequences whose support is greater than a user predefined threshold, *min_sup*.

The absolute and relative support of a document $d = \{S1, S2, \ldots, Sn\}$, where Si is a sequence representing a

paragraph in *d*. Let *P* be a sequence. We call *P* a sequential pattern of *d* if there is a *Si* ∈*d* such that *P* ⊆*Si* . The absolute support of *P*, denoted as *suppa*(*P*) = |{ *S* | *S* ∈*d* ∧ *P* ⊆*S* }|, is the number of occurrences of *P* in *d*. The relative support of *P* is the fraction of paragraphs that contain *P* in document *d*, denoted as *suppr*(*P*) = *suppa*(*P*) / | *d* |. For example, the sequential pattern *P* = <A, B, C> in the sample database (Table 1) has *suppa*(*P*) = 2 and *suppr*(*P*) = 0.5 for the document in Table-1.

The frequent sequential patterns *P* is called frequent sequential pattern if *suppr*(*p*) is greater than or equal to a minimum support *min_sup*. For example, let *min_sup*= 0.75 for the document shown in Table-1; we can obtain four frequent sequential patterns: <B, C>, <A>, <B>, and <C> since their relative supports are not less than *min_sup*.

The purpose of using *min_sup*in our model is to reduce the number of patterns discovered in a large document. Otherwise these patterns with lower relative support will increase the burden of the training. Removing less significant patterns will save much computation time without affecting the performance very much. An example is given in table 2 to show that only a small amount of frequent sequential patterns left after using *min_sup*.

A frequent sequential pattern *P* is a *maximal sequential pattern* if there exists no frequent sequential pattern *P′* such that *P* ⊂*P′* and *suppa*(*P*) = *suppa*(*P′*).

The length of sequential pattern *P*, denoted as *len*(*P*), indicates the number of words (or terms) contained in *P*. A sequential pattern which contains *n* terms can be denoted in short as *nTerms*pattern.

For instance, given pattern *P* = <B, C>, we have *len*(*P*) = 2, and *P* is a *2Terms* pattern. Although a sequential pattern consists of several terms (words), *1Term* pattern is a sort of special *nTerms*pattern in this study.

4. PATTERN CLASSIFICATION MODEL

In this paper, we present a new pattern-based classification model PCM (Pattern Classification Model) for the representation of text documents. Pattern classification follows a tree-like structure that illustrates the relationship between patterns extracted from a text collection.

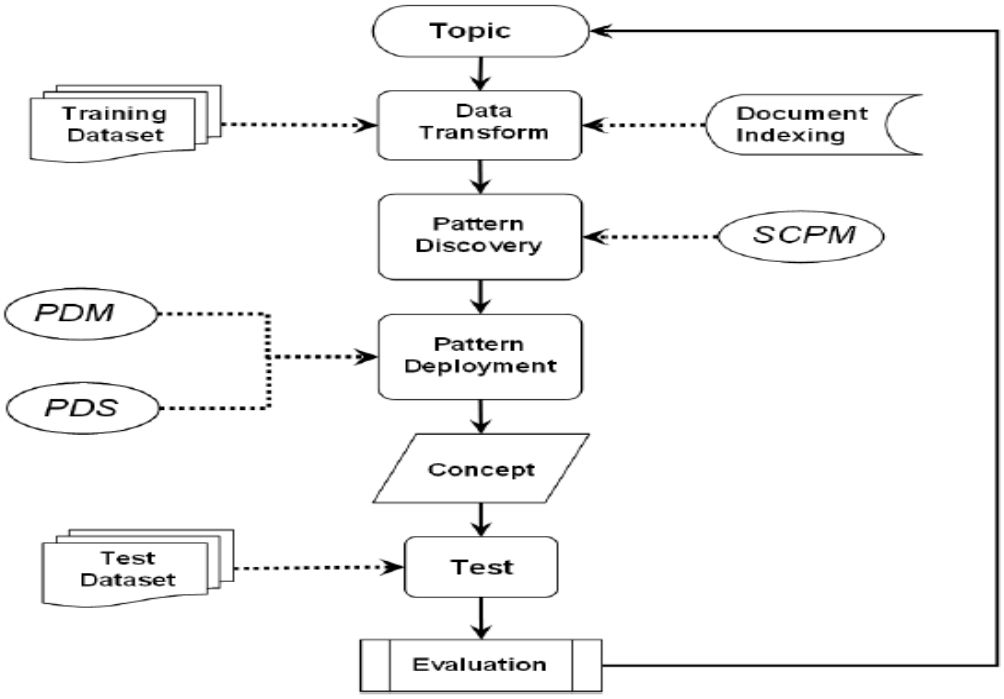

Figure-2. Experimental procedure for pattern classification methods [1].

The arrow indicates the sub-sequence relation between patterns. For example, pattern <A, B> is a sub-sequence of pattern <A, B, C>, and pattern <B> is a sub-sequence of pattern <B, C>. The root of the tree in the bottom level represents one of the longest patterns (i.e., maximum sequential patterns). Once the tree is constructed, we can easily find the relationship between patterns. The next step is to prune the meaningless patterns in the pattern taxonomy.

A frequent sequential pattern *P*1 is a ***closed pattern*** of *P*2 if *P*2 is a frequent sequential pattern, *P*1⊆*P*2, and *suppa*(*P*1) – *suppa*(*P*2) = 0.

The followings are the definitions of two operations used in the algorithm: sequence extension and p-projected database.

Algorithm: 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* frequentsequential patterns.)
Output: a set of frequent sequential patterns *SP*.
Method:
1) SP = SP – {Pa ∈SP |∃Pb∈PL such thatlen(Pa) = len(Pb) -1 ⊂Pa ⊂Pb∧
suppa(Pa) = suppa(Pb) } /* pruning */
2) SP ←SP ∪PL /* add found patterns */
3) PL′← {φ} /* PL′ : set of (n+1)Termsfrequent sequential patterns */
4) **foreach**pattern p in PL **do begin**
5) generatep-projected database PD
6) **foreach**frequent term t in PD **do begin**
7) P′←p⋈t /* P′: set of (n+1)Termssequential candidates */
8) **if**suppr(P′) ≥ min_sup**then**
9) PL′←PL′∪P′
10) **end if**
11) **end for**
12) **end for**
13) **if**|PL′ | = 0 **then**
14) **return /*** no more patterns found */
15) **else**
16) **call** SPMining(PL′, min_sup)
17) **end if**
18) output frequent sequential patterns in SP

The objective of pruning phase is to eliminate the meaningless patterns. As can be seen in the Figure 1, pattern<A, B> is a closed pattern of <A, B, C>. That means they always appear in the same paragraph. Therefore, the shorter one (i.e., pattern <A,B>) is negligible and is considered as a meaningless pattern. We keep the longer one since it is more meaningful and carry more information than the shorter one. Thus, after the pruning phase, only the significant patterns remain in the pattern classification.

5. CLOSED SEQUENTIAL PATTERNS

A sequential pattern s =<t1, . . . ,tr>(ti∈ T) is an ordered list of terms. A sequence s1 =<x1, . . . , xi> is a subsequence of another sequence s2 =<y1, . . . ,yj>, denoted by s1 ⊆ s2, if f∃j1, . . . ; jy such that $1 \leq j_1 < j_2 \ldots < j_y \leq j$ and $x_1 = y_{j1}$; $x_2 = y_{j2}$,. . . , $x_i = yj_y$. Given s1 ⊆ s2, we usually say s1 is a sub pattern of s2, and s2 is a super pattern of s1. In the following, we simply say patterns for sequential patterns. Given a pattern (an ordered term set) X in document d, χ is still used to denote the covering set of X, which includes all paragraphs ps∈ P(d) such that X ⊆ps, i.e., χ ={ps|ps∈ PS(d), X ⊆ps}. Its absolute support is the number of occurrences of X in PS(d), that is supa(X)= |χ |. Its relative support is the fraction of the paragraphs that contain the pattern, that is, supr(X) == | χ |/|ps(d)|.

A sequential pattern X is called frequent pattern if its relative support (or absolute support) ≥min_sup, a minimum support. The property of closed patterns can be used to define closed sequential patterns. A frequent sequential pattern X is called closed if not ∃ any super-patternX1 of X such that supa(X1) =supa(X).

6. 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 [17]. 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, stop word removal is applied and the Porter algorithm [13] is selected for suffix stripping. Terms with term frequency equaling to one are discarded.

7. MEASURES

Several standard measures based on precision and recall is 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 order to assess the effect involving both precision and recall, another criterion that can be used for experimental evaluation is F_-measure [20], which combines precision and recall and can be defined by the following equation:

$$F_{\beta}\text{-measure} = (\beta^2 + 1) * \text{precision} * \text{recall}/(\beta^2 * \text{precision} + \text{recall}); \quad (1)$$

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 [11]. A value β = 1 is adopted in our experiments meaning that it attributes equal importance to precision and recall. When β = 1, the measure is expressed as:

$$F1 = 2 * \text{precision\_recall}/(\text{precision} + \text{recall}) \quad (2)$$

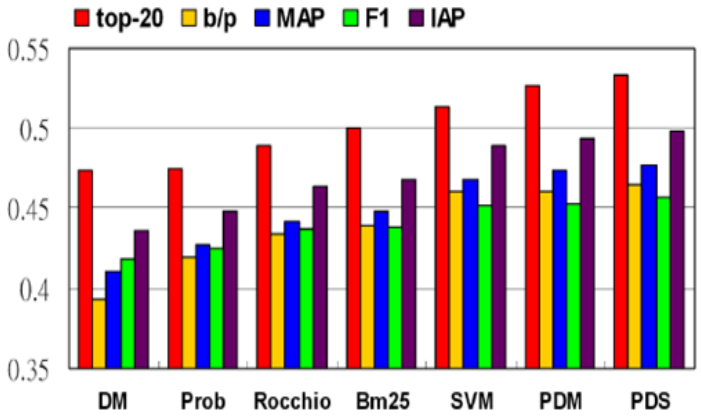

Figure-3. Comparison of all measures on 110 topics.

The value of $F_{\beta=1}$ is equivalent to the b/p when precision equals to recall. However, the b/p cannot be compared directly to the $F_{\beta=1}$ value since the latter is given a higher score than that of the former [54]. It has also been stated in [10] that the $F_{\beta=1}$ measure is greater or equal to the value of b/p.

8. 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 that are of two classes: the concept-based model and term-based methods.

CONCEPT-BASED MODELS

A new concept-based model was presented in [4, 5] and [4, 6], 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 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:

$$Weight(c)=tf_{weight(c)}+ctf_{weight(c)} \quad (3)$$

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 {hits} and its argument "the ball" is denoted as {the; ball}.

The first step is to find all of the concepts in the positive documents of the training set, where verbs are extracted from Prop Bank data set. The second step is to use the deploying approach to evaluate the weights of terms based on their appearances in these discovery concepts. Unlike the proposed model, which uses 4,000 features at most, the concept-based model uses all features for each topic. Let CP_i be the set of concepts in $d_i \in D^+$. To synthesize both tf and ctf of concepts in all positive documents, we use the following equation to evaluate term weights.

$$W(t) = \sum_{i=1}^{|D+|}(|\{c|c\in CPi, t\in c\}|)/\sum_{c\in cpi}|c \quad (4)$$

For all $t \in T$

We also designed another kind of the concept-based model, called CBM Pattern Matching, which evaluates a document d's relevance by accumulating the weights of concepts that appear in d as follows:

$$Weight(d) = \sum_{cd} weight(c). \quad (5)$$

TERM BASED METHODS

There are many classic term-based approaches. The Rocchio algorithm [14], which has been widely adopted in information retrieval, can build text representation of a training set using a Centroid $\vec{c}$ as follows:

$$\vec{c} = \alpha \frac{1}{|D|} \sum_{\vec{d}\in D+}^{n} \frac{\vec{d}}{||\vec{d}||} - \beta \frac{1}{|D-|} \sum_{\vec{d}\in D-} \frac{\vec{d}}{||\vec{d}||} \quad (6)$$

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

Probabilistic methods (Prob) are well-known term-based approaches. The following is the best one:

$$W(t) = \log(((r+0.5)/R-r+0.5)/((n-r+0.5)/(N-n)-(R-r)+0.5)) \quad (7)$$

where N and R are the total number of documents and the number of positive documents in the training set, respectively; n is the number of documents which contain

t; and r is the number of positive documents which contain t.

In addition, TFIDF is also widely used. The term t can be weighted by W(t) = TF(d, t) * IDF(t), where term frequency TF(d, t) is the number of times that term t occurs in document d(d ∈ D) (D is a set of documents in the data set); DF(t) is the document frequency which is the number of documents that contain term t; and IDF(t) is the inverse document frequency.

Another well-known term-based model is the BM25 approach, which is basically considered the state-of-the-art baseline in IR [35]. The weight of a term t can be estimated by using the following function:

$$W(t) = \frac{TF.(k1+1)}{k1.((1-b)+b\left(\frac{DL}{AVDL}\right)+TF} .log \frac{(r+0.5)/(n-r+0.5)}{(R-r+0.5)/(N-n-R+r+0.5)} \quad (8)$$

where TF is the term frequency; k1 and b are the parameters; DL and AV DL are the document length and average document length. The values of k1 and b are set as 1.2 and 0.75, respectively, according to the suggestion in [17] and [18].

The SVM model is also a well-known learning method introduced by Cortes and Vapnik [8]. Since the works of Joachim's [15], [16], researchers have successfully applied SVM to many related tasks and presented some convincing results [5], [6], [17]. The decision function in SVM is defined as:

$$h(x) = sign(W \cdot x + b) = \begin{cases} +1, & \text{if } (W \cdot x + b) > 0, \\ -1, & \text{else,} \end{cases} \quad (9)$$

$$h(x) = sign(W.x + b) = \{+1 \text{ if } W.x + b > 0, \quad -1 \text{ else,} \quad (10)$$

where x is the input space; b ∈ R is a threshold and

$$W = \sum_{i=1}^{l} y_i \alpha_i x_i$$

the given training data $(x_i, y_i), \ldots (x_l, y_l)$; where xi ∈Rn and yi equals +1 (-1), if document xi is labeled positive (negative). $\alpha_i \in R$ is the weight of the training example xi and satisfies the following constraints

$$\forall i : \alpha_i \geq 0, \quad \text{and} \quad \sum_{i=1}^{l} \alpha_i y_i = 0. \quad (11)$$

Since all positive documents are treated equally before the process of document evaluation, the value of _i is set as 1.0 for all of the positive documents and thus the _i value for the negative documents can be determined by using (13).

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 vector of d and ~c is the concept of the topic.

$$weight(d) = \sum_{t \in T} W(t)\tau(t,d). \quad (12)$$

Table-1. Number of relevant documents (#r) and total number of documents (#d) by each topic in the RCV1 training dataset.

No.	#r	#d	No.	#r	#d	No.	#r	#d	No.	#r	#d
r101	7	23	r126	19	29	r151	6	49	r176	5	57
r102	135	199	r127	5	32	r152	5	55	r177	25	45
r103	14	64	r128	4	51	r153	10	18	r178	3	43
r104	120	194	r129	17	72	r154	6	52	r179	5	57
r105	16	37	r130	3	24	r155	11	74	r180	5	61
r106	4	44	r131	4	31	r156	6	37	r181	4	64
r107	3	61	r132	7	103	r157	3	42	r182	19	36
r108	3	53	r133	5	47	r158	5	79	r183	25	55
r109	20	40	r134	5	31	r159	21	62	r184	9	48
r110	5	91	r135	14	29	r160	15	36	r185	26	52
r111	3	52	r136	8	46	r161	5	52	r186	20	38
r112	6	57	r137	3	50	r162	6	27	r187	7	48
r113	12	68	r138	7	98	r163	4	29	r188	3	30
r114	5	25	r139	3	21	r164	21	64	r189	12	56
r115	3	46	r140	11	59	r165	7	53	r190	13	42
r116	16	46	r141	24	56	r166	8	39	r191	5	43
r117	3	13	r142	4	28	r167	5	63	r192	3	40
r118	3	32	r143	4	52	r168	32	43	r193	5	64
r119	4	26	r144	6	50	r169	5	35	r194	31	80
r120	9	54	r145	5	95	r170	16	79	r195	8	36
r121	14	81	r146	13	32	r171	7	48	r196	5	61
r122	15	70	r147	6	62	r172	10	78	r197	22	34
r123	3	51	r148	12	33	r173	27	35	r198	3	29
r124	6	33	r149	5	26	r174	5	44	r199	21	40
r125	12	36	r150	4	51	r175	37	37	r200	7	34

The most important information revealed in this table 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 outperforms CBM Pattern Matching and CBM in the five measures. CBM outperforms all other models for the first 50 topics. For the time complexity in the testing phase, all models take $O(|T| * |d|)$ 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.

Table-2. Comparing PTM with data mining-based methods on RCV1 topics r101 to r150.

Method	Pattern type	#Patterns	Runtime (Sec)	b/p
SPM	Sequential Pattern	126.310	5.308	0.343
SCPM	Sequential Closed Pattern	38.588	4.653	0.353
NSPM	Frequent Itemset	340.142	14.502	0.352
NSCPM	Frequent Closed Itemset	34.794	7.122	0.346
3Gram	nGram	88.991	4.092	0.342
PCM(PCS)	Pattern Classification	8.027	1.602	0.521

Figure-3. Number of patterns discovered using SPM with different constraints on 10 RCV1 topics.

Topic	#doc	#frequent sequential patterns		
		min_sup =0	min_sup =0.2	min_sup=0.2 &pruning
110	491	9,977	5,252	5,252
120	415	5,395	3,933	2,959
130	307	4,128	1,948	1,845
140	432	16,688	4,007	3,227
150	371	8,492	5,022	3,646
160	199	4,032	2,060	1,929
170	507	12,239	6,649	4,745
180	426	26,098	2,023	1,794
190	337	4,382	2,780	2,085
200	277	3,227	1,996	1,251
Total	3,762	94,658	36,202	28,733
Avg.b/p		0.409	0.406	0.443

Table-4. Precisions of top 20 returned documents on10 RCV1 topics.

Topic	TFIDF	Prob	SPM	SCPM	PCM(PCS)
r110	0.15	0.30	0.45	0.65	0.50
r120	0.45	0.30	0.80	0.60	0.65
r130	0.05	0.05	0.10	0.25	0.25
r140	0.35	0.30	0.45	0.10	0.65
r150	0.15	0.01	0.10	0.10	0.20
r160	0.90	1.00	0.95	1.00	1.00
r170	0.30	0.30	0.55	0.60	0.50
r180	0.70	0.70	0.65	0.65	0.65
r190	0.75	0.60	0.80	0.80	0.95
r200	0.20	0.50	0.20	0.40	0.70
top-20	0.400	0.406	0.505	0.515	0.605

Table-5. Results of pattern deploying methods compared with others on the first 50 topics.

	Prob	DM	Rocchio	PDM	PDS
Top-20	0.407	0.416	0.416	0.470	0.490
b/p	0.381	0.353	0.392	0.427	0.431
MAP	0.379	0.364	0.391	0.435	0.441
$F_{\beta=1}$	0.396	0.390	0.408	0.435	0.440
IAP	0.402	0.392	0.418	0.458	0.465

Table-6. Results of pattern deploying methods compared with others on the last 50 topics.

	Prob	**DM**	**Rocchio**	**PDM**	**PDS**
Top-20	0.542	0.540	0.562	**0.583**	0.576
b/p	0.457	0.434	0.476	0.492	0.498
MAP	0.476	0.456	0.492	0.512	0.513
$F_{\beta=1}$	0.454	0.445	0.465	0.471	0.473
IAP	0.493	0.479	0.508	0.529	0.531

Table-7. Results of pattern deploying methods compared with others on all topics.

	Prob	**DM**	**Rocchio**	**PDM**	**PDS**
Top-20	0.475	0.473	0.489	0.527	0.533
b/p	0.419	0.394	0.434	0.460	0.464
MAP	0.427	0.410	0.442	0.473	0.477
$F_{\beta=1}$	0.425	0.417	0.436	0.453	0.457
IAP	0.447	0.435	0.463	0.493	0.498

Table-8. Accumulated number of patterns found during pattern discovering.

		Topic	
	First 50	Last 50	All
Prob	32,760	37,418	70,178
DM	38,588	39,317	77,905
Rocchio	32,760	37,418	70,178
PDM,PDS,PCM	8,027	11,838	19,865

Table-9. The list of methods used for evaluations.

Method	**Description**	**Algorithm**
PCM	Proposed method equipped with PDS and IPE	IPE
Sequential ptns.	Data mining method using frequent sequential patterns	SPM
Sequential closed ptns.	Data mining method using frequent sequential closed patterns	SCPM
Freq. Itemsets	Data mining method using frequent item sets	NSPM
Freq. closed Itemsets	Data mining method using frequent closed itemsets	NSCPM
nGram	nGram method with n=3	3Gram
Rocchio	Rocchio method	Equation 6.5, $\alpha=1$, $\beta=0$
Prob	Probabilistic method	Equations 6.11, $\eta=0.5$
TFIDF	TFIDF method	TFIDF
BM25	Probabilistic method	Equation 6,K_1=1.2,b=0.75
SVM	Support vector machines method	Equation 6, b=0

Table-10. Comparison of pattern deploying and pattern evolving methods used by PTM on all topics.

	PDS	PDM	$DPE_{\mu=3}$	$DPE_{\mu=5}$	$DPE_{\mu=7}$	IPE
Top-20	0.5330	0.5265	0.5280	**0.5285**	**0.5275**	**0.5360**
b/p	0.4643	0.4598	0.4507	0.4507	0.4516	**0.4632**
MAP	0.4768	0.4734	0.4649	0.4652	0.4653	**0.4770**
$F_{\beta=1}$	0.4565	0.4528	0.4519	0.4520	0.4520	**0.4570**
IAP	0.4982	0.4932	0.4861	0.4867	0.4867	**0.4994**

Table-11. Comparison of all methods on the first 50 topics.

Method	Top-20	b/p	MAP	$F_{\beta=1}$	IAP
PCM(IPE)	**0.501**	**0.449**	**0.461**	**0.460**	**0.486**
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
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	0434

9. PCM VERSUS OTHER MODELS

The total number of patterns is estimated by accumulating the number for each topic. As a result, the figure shows PCM (IPE) is the method that utilizes the least amount of patterns for concept learning compared to others. This is because the efficient scheme of pattern pruning is applied to the PCM (IPE) method. Nevertheless, the classic methods such as Rocchio, Prob, and TFIDF adopt terms as patterns in the feature space; they use much more patterns than the proposed PTM (IPE) method and slightly less than the sequential closed pattern mining method. Particularly, nGram and the concept-based models are the methods with the lowest performance which requires more than 15, 000 patterns for concept learning. In addition, the total number of patterns obtained based on the first 50 topics is almost the same as the number obtained based on the last 50 topics for all methods except PTM (IPE).

Based on the first topics group (r101 r150) for PCM (IPE) is less than that based on the other group (r151 r200)? This can be explained in that the high proportion of closed patterns is obtained by using PTM (IPE) based on the first topics group.

A further investigation in the comparison of PCM (IPE) and TFIDF in top-20 precision on all RCV1 topics is depicted. It is obvious that PCM (IPE) is superior to TFIDF as it can be seen that positive results distribute over all topics, especially for the first 50 topics. Another observation is the scores on the first 50 topics are better than those on the last fifty. That is because of the different ways of generating these two sets of topics, which has been mentioned before. The interesting behavior is that there are a few topics where TFIDF outperforms PTM. After further investigation, we found a similar characteristic of these topics in that there are only a few positive examples available in these topics. For example, topic r157, which is the worst case for PCM (IPE) compared to TFIDF, has only three positive documents available. Note that the average number of positive documents for each topic is over 12. The similar behaviors are found in topics r134 and r144.

The plotting of precisions on 11 standard points for PCM (IPE) and pattern mining based methods on the first 50 topics is illustrated in Figure-9. The result supports the superiority of the PCM (IPE) method and highlights the importance of the adoption of proper pattern deploying and pattern evolving methods to a pattern-based knowledge discovery system. Comparing their performance at the first few points around the low-recall area, it is also found that the points for pattern mining methods drop rapidly as the recall value rises and then keep a relatively gradual slope from the mid recall period to the end. All four pattern mining methods achieve similar results. However, the plotting curve for PTM (IPE) is much smoother than those for pattern mining methods as there is no severe fluctuation on it. Another observation

on this figure is that the pattern mining-based methods however perform well at the point where recall is close to zero, despite the overall unpromising results they have. Accordingly, we can conclude that the pattern mining-based methods can improve the performance in the low-recall situation.

As mentioned before, data mining-based methods can perform well at the low recall area, which can explain why nGram has better results at this point. However, the scores for the nGram method drop rapidly at the following couple of points. During that period, SVM, BM25, Rocchio, and Prob methods transcend the nGram method and keep the superiority until the last point where recall equals to 1. There is no doubt that the lowest performance is produced by the TFIDF method, which outperforms the nGram method only at the last fewer call points.

In addition, the Prob method is superior to the nGram method, but inferior to the Rocchio method. The overall performance of Rocchio is better than that for the Prob method which corresponds to the finding in [50].

In summary, the proposed approach PCM (IPE) achieves an outstanding performance for text mining by comparing with the up-to-date data mining-based methods, the concept models, and the well-known term-based methods, including the state-of-the-art BM25 and SVM models. The results show the PCM (IPE) model can produce encouraging gains in effectiveness, in particular over the SVM and CBM models. These results strongly support Hypothesis H1. The promising results can be explained in that the use of the deploying method is promising (Hypothesis H2 is also supported) for solving the misinterpretation problem because it can combine well with the advantages of terms and discovered patterns or concepts. Moreover, the inner pattern deploying strategy provides an effective evaluation for reducing the side effects of noisy patterns because the estimation of term weights in the term space is based on not only terms' statistical properties but also patterns' associations in the corresponding pattern taxonomies.

10. CONCLUSIONS

The existed data mining techniques like association rule mining, frequent item set mining, sequential pattern mining, maximum pattern mining, and closed pattern mining were not given proper efficiency for 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). Our experiments proved all patterns are not useful which cause low 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] K. Aas and L. Eikvil. 1999. Text Categorisation: A Survey. Technical Report Raport NR 941, Norwegian Computing Center.

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

[3] H. Ahonen, O. Heinonen, M. Klemettinen and A.I. Verkamo. 1998. 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.

[4] R. Baeza-Yates and B. Ribeiro-Neto. 1999. Modern Information Retrieval. Addison Wesley.

[5] N. Cancedda, N. Cesa-Bianchi, A. Conconi, and C. Gentile. 2002. Kernel Methods for Document Filtering. TREC, trec.nist.gov/pubs/trec11/papers/kermit.ps.gz.

[6] N. Cancedda, E. Gaussier, C. Goutte and J.-M. Renders. 2003. Word-Sequence Kernels. J. Machine Learning Research. 3: 1059-1082.

[7] M.F. Caropreso, S. Matwin, and F. Sebastiani. 2000. Statistical Phrasesin Automated Text Categorization. Technical Report IEI-B4-07-2000, Instituto di Elaborazionedell' Informazione.

[8] C. Cortes and V. Vapnik. 1995. Support-Vector Networks. Machine Learning. 20(3): 273-297.

[9] S.T. Dumais. 1991. Improving the Retrieval of Information from External Sources. Behavior Research Methods, Instruments, and Computers. 23(2): 229-236.

[10] J. Han and K.C.-C. Chang. 2000. Data Mining for Web Intelligence. Computer. 35(11): 64-70.

[11] J. Han, J. Pei, and Y. Yin. 2000. Mining Frequent Patterns without Candidate Generation. Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '00), pp. 1-12.

[12] Y. Huang and S. Lin. 2003. Mining Sequential Patterns Using Graph Search Techniques. Proc. 27th Ann. Int'l Computer Software and Applications Conf. pp. 4-9.

[13] N. Jindal and B. Liu. 2006. Identifying Comparative Sentences in Text Documents. Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '06). pp. 244-251.

[14] T. Joachims. 1997. A Probabilistic Analysis of the Rocchio Algorithm with tfidf for Text Categorization. Proc. 14th Int'l Conf. Machine Learning (ICML '97). pp. 143-151.

[15] T. Joachims. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features," Proc. European Conf. Machine Learning (ICML '98), pp. 137-142.

[16] T. Joachims. 1999. Transductive Inference for Text Classification Using Support Vector Machines. Proc. 16th Int'l Conf. Machine Learning (ICML '99). pp. 200-209.

[17] W. Lam, M.E. Ruiz and P. Srinivasan. 1999. Automatic Text Categorization and Its Application to Text Retrieval. IEEE Trans. Knowledge and Data Eng. 11(6): 865-879.

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

[19] D.D. Lewis. 1992. Feature Selection and Feature Extraction for Text Categorization. Proc. Workshop Speech and Natural Language. pp. 212-217.

[20] D.D. Lewis. 1995. Evaluating and Optimizing Automous Text Classification Systems. Proc. 18th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '95). pp. 246-254.

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

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

[23] Y. Li, C. Zhang, and J.R. Swan. 2000. An Information Filtering Modelon the Web and Its Application in Jobagent. Knowledge-Based Systems. 13(5): 285-296.

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