Pattern Discovery Text Mining for Document Classification

Mustafa M. Shaikh
IT Department
Dr.D.Y.Patil Institute of Engineering and Technology, Ambi, Pune

Ashwini A. Pawar
IT Department
Dr.D.Y.Patil Institute of Engineering and Technology, Ambi, Pune

Vibha B. Lahane
Asst. Professor, IT Department
Dr.D.Y.Patil Institute of Engineering and Technology, Ambi, Pune

ABSTRACT
In text documents data mining techniques have been discovered for mining useful patterns. But there are some questions, how to properly use and update discovered patterns is still an open research issue, specifically in the text mining. Therefore most existing text mining methods have used term-based approaches but they all suffer from the problems of polysemy (multiple meaning word) and synonymy(same meaning word).This is the literature survey paper with proposed system develops innovative and successful pattern-based technique which contains the processes of pattern taxonomy, pattern deploying and gradually developing pattern ,to improve the effectiveness of using and updating researched patterns for finding applicable and interesting data with effectual patterns as per the users requirements. In this paper user is also getting the meaningful information without wrong meaning problem.

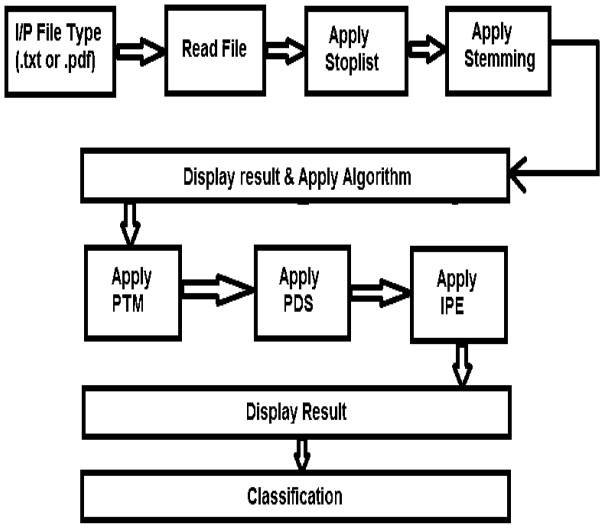
General Terms
Text mining, information filtering, Pattern Mining, Data Mining, pattern evolving, Text Classification.

Keywords
D-matrix ,Pattern Taxonomy

1. INTRODUCTION
Text mining is the discovery of interesting information in word documents. It is a stimulating issue to find precise information in text documents to help users to find what they want. Many applications, such as market analysis and business managing, can profit by the use of the data and facts extracted from a large amount of data. Knowledge discovery can be effectively use and update discovered patterns and apply it to field of text mining .Data mining is therefore an essential step in the process of knowledge discovery in databases, which means data mining is having all methods of knowledge discovery process and presenting modeling phase that is application of methods and algorithm for calculation of search pattern or models. These techniques include association rule mining(ASM), frequent item set mining(FIM), sequential pattern mining(SPM), maximum pattern mining(MPM) and closed pattern mining(CPM). Utmost of them are projected for the determination of developing efficient mining algorithms to find particular patterns within a equitable and tolerable time structure. With a great number of patterns engendered by using the data mining methods, how to effectively exploit these patterns is still an exposed research issue.

2. BASIC CONCEPTS
Taxonomy of entities for search engines is designed to improve significance [17]in perpendicular search. Taxonomies of objects are trees whose nodes are labeled with entities which expect to occur in a web exploring request. These trees are used to compare keywords from search query with the keywords from answers (or snippets).Taxonomies, thesauri and concept hierarchies are crucial components for many applications of Natural Language processing, Information Retrieval and information management. Though, construction, regulation and handling taxonomies and ontologies is rather costly since a lot of manual processes are essential. A number of studies projected the programmed construction of taxonomies based on verbal resources and or statistical machine learning Web mining is one of the methods to form search engine taxonomies for net search. The taxonomy building procedure starts from the kernel objects and mines accessible source areas for new objects linked with these seed objects. New objects are molded by put on the machine learning to the present net search results for standing objects to form harmonies among them. These unity words then form parameters of present objects, and are revolved into new objects at the next learning repetition.

3. SYSTEM ARCHITECTURE



4. PROPOSED WORK
Research project selection is an important task for government and private research funding organizations. When a great number of research applications are received, it is mutual to cluster them conferring to their resemblances in research domains. The clustered proposals are then allotted to the suitable specialists for peer analysis. Present procedures for clustering proposals are built on manual toning of related research domain areas and/or keywords.[13] However, the exact research domain areas of the offers cannot often be exactly nominated by the candidates due to their particular views and possible misapprehensions. So, rich info in the

applications' full text can be used efficiently. Text-mining approaches have been projected to solve the problem by automatically categorizing text documents, largely in English.

4.1 Preprocessing

All words passes to pre-processing step. Inappropriate terms are removed there. This procedure is also called as **tokenization** procedure. It contains two types of processes such as stop list elimination, stem word elimination.

A). Stop List Elimination: Stop words are words which are cleaned out proceeding to, or afterward, treating of natural language data. They typically comprise prepositions, articles, and so on. There is no specific list of stop words for all applications and these stop words are controlled by the human but not automated. It saves the system assets. Stop word has list of arguments. That are considered as inappropriate and then it is eliminated .It consists of (a, an, the) articles, (for, in, at, etc.) preposition, etc.

B). Stem word removal: Stemming is the process for reducing inflected (or sometimes derived) words to their stem base or root form. It generally a written word forms. In this preprocess the text documents have to be processed using the Porter stemmer. It removes the Suffix's of the words these words are useful in the text mining for clustering the text documents in the text mining process users collects the documents and each documents are composed into the set of terms or words the words having stem have a same meaning in stem process the suffixes of the words, singular and plural words are considered into a one single word for meaning full text clustering process.

4.2 Pattern Taxonomy Model

Users assume that all documents are split into paragraphs. So a given document *A* yields a set of paragraphs PS(A). Let B be a teaching set of docs, which contains a set of docs, B;· Let C= {c_1, c_2, …,c_n} be a set of terms (or keywords) which can be extracted from the set of documents, B

T1:A set of paragraphs

Paragraphs	Terms
Ap_1	c_1 c_2
Ap_2	c_3 c_4 c_6
Ap_3	c_3 c_4 c_5 c_6
Ap_4	c_3 c_4 c_5 c_6
Ap_5	c_1 c_2 c_6 c_7
Ap_6	c_1 c_2 c_6 c_7

A).Frequent and Closed Patterns-

Given [1] a term set D in document d , $\ulcorner D \urcorner$ is used to denote the covering set of D for A, which includes all paragraphs Ap PS(A) such that D⊆Ap, i.e.,

$$\lceil D \rceil = \{\text{Ap} | \text{Ap } PS(A), D \; Ap\}.$$

Its absolute support is the number of occurrences of D in PS(A),that is $sup_a(D)=|\ \ulcorner D \urcorner\ |$.Its relative support is the fraction of the paragraphs that have the pattern, that is

$$sup_r(D)=\frac{|\ \ulcorner D \urcorner\ |}{|PS(A)|}$$

A termset D is called frequent pattern if its sup_r(or sup_a)≥min_sup, minimum support.

Table 1 lists a set of paragraphs for a given document A, where PS(A)={Ap_1,Ap_2, ….,Ap_6} and duplicate terms were removed.

T2: Frequent patterns and covering sets

Frequent patterns	Covering sets
{c_3, c_4, c_6}	{Ap_2,Ap_3,Ap_4}
{c_3, c_4}	{Ap_2,Ap_3,Ap_4}
{c_3, c_6}	{Ap_2,Ap_3,Ap_4}
{c_4, c_6}	{Ap_2,Ap_3,Ap_4}
{c_3}	{Ap_2,Ap_3,Ap_4}
{c_4}	{Ap_2,Ap_3,Ap_4}
{c_1, c_2}	{Ap_1,Ap_5,Ap_6}
{c_1}	{Ap_2,Ap_3,Ap_4}
{c_2}	{Ap_2,Ap_3,Ap_4}
{c_6}	{Ap_2,Ap_3,Ap_4 ,Ap_5,Ap_6}

Let min_sup=50%, users can obtain ten frequent patterns in Table 1 using the above explanations. T2 illuminates the ten frequent patterns and their covering sets. Not all repeated patterns in T2 are beneficial. For example, pattern {c_3, c_4} always occurs with term c_6 in paragraphs, i.e., the shorter pattern, {c_3, c_4}, is always a part of the larger pattern, {c_3, c_4, c_6}, in all of the paragraphs. Therefore, users consider that the smaller one, {c_3, c_4}, is a noise pattern and expect to keep the larger pattern, {c_3, c_4, c_6}, only. Given a term set D, its covering set | $\ulcorner D \urcorner$ | is a subset of paragraphs. Similarly, given a set of paragraphs E ⊆ PS (A), users can define its term set, which satisfies

termset(E)={$c|\forall\ Ap \in E \rightarrow c \in Ap$}

The closure of is defined as follows:

$$\text{Cls(D)=termset}(\ulcorner D \urcorner).$$

A pattern D (also a termset) is called closed if and only if

D=Cls(D).

Let D be a closed pattern. Users can prove that

$$sup_a(D1)<sup_a(D) \qquad (1)$$

for all patterns D1 ⊃ D; otherwise, if $sup_a(D1)= sup_a(D)$,

users have

$$\lceil D1 \rceil = \lceil D \rceil$$

Where $sup_a(D1)$ and $sup_a(D)$ are the absolute support of pattern D1 and D, respectively.

Users also have

$$\text{Cls(D)} = \text{termset}(\lceil D \rceil) = \text{termset}(\lceil D1 \rceil) \supseteq \text{D1} \supset \text{D},$$

that is, Cls(D) ≠D.

B). Pattern Taxonomy-
Patterns can be structured into a taxonomy by using the is-a (or subset) relative. For the example of T1, where users have demonstrated a set of paragraphs of a document, and the exposed 10 repeated patterns in T2 if assuming min_sup = 50%. There are, however, only threeclosed patterns in this example. They are $<c_3,c_4, c_6>$, $<c_1, c_2>$, and $<c_6>$.

Fig. 1 illustrates an example of the pattern taxonomy for the frequent patterns in T2, where the nodes denote repeated 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 $\{c_6\}$ would become a direct sub-pattern of $\{c_3,c_4,c_6\}$,after pruning non-closed patterns. Smaller patterns in the taxonomy, for example pattern $\{c_6\}$ (see Fig. 1) are usually more general because they could be used frequently in both positive and documents; and larger patterns, for example pattern $\{c_3,c_4,c_6\}$ in the taxonomy are usually more specific since they may be used only in confident documents. The semantic info will be used in the pattern taxonomy to develop the performance of using closed patterns in text mining.

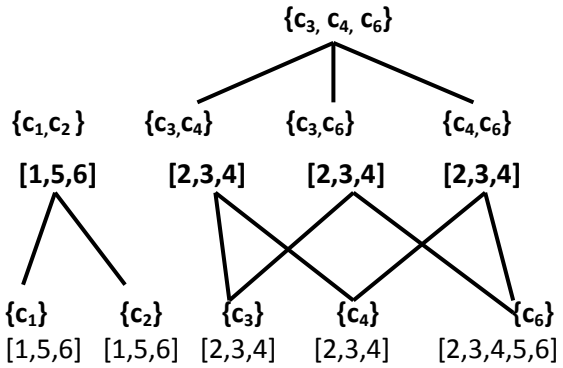


Fig 1: Pattern taxonomy

C). Closed Sequential Patterns
A [2]sequential pattern $s = <c_1,. . . ,c_r>$ ($ci \in C$) is an ordered list of terms. A sequence $s_1 = <d_1, . . . ,d_i>$is a subsequence of another sequence $s_2=<e_1,. . . ,e_j>$, denoted by $s_1 \hat{o} \; s_2$, if $\exists j_1, . . . ,j_y$ such that $1 \leq j_1 < j_2 . . . < j_e \leq j$ and $d_1 = e_{j1}, d_2=e_{j2}, . . . , d_i = e_{je}$. Given s1 $\hat{o}$ s2, users usually say s1 is a sub pattern of s_2, and s_2 is a super pattern of s_1. In the following, users simply say patterns for sequential patterns. Given a pattern (an ordered termset) D in document A, $\lceil D \rceil$ is still used to denote the covering set of D, which includes all paragraphs $ps \in PS(A)$ such that $D \hat{o} \; ps$, i.e., $\lceil D \rceil = \{ ps \mid ps \in PS(A), D \hat{o} \; ps \}$.Its absolute support is the number of occurrences of D in PS(A), that is $sup_a(D) = | \lceil D \rceil |$. Its relative [3]support is the fraction of the paragraphs that contain the pattern, that is,

$$sup_r(D) = \frac{| \lceil D \rceil |}{|PS(A)|}.$$

A sequential pattern D is called frequent pattern if its relative support (or absolute support) _ min sup, a minimum support. The property of closed patterns (see eq. (1)) can be used to define closed sequential patterns. A frequent sequential pattern D is called closed if not 9 any superpatternD1 of D such that $sup_a(D1) = sup_a(D)$.

4.3 Pattern Deploying Method

In demand to use the semantic info in the pattern taxonomy to develop the performance of closed patterns in text mining, users need to interpret discovered patterns by summarizing them as d-patterns (see the definition below) in order to accurately evaluate term weights (supports).

The rationale 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 methods. In the term-based methods, the estimate of term weights is based on the distribution of terms in documents. In this ,terms are biased conferring to their forms in exposed closed patterns.

A). Representation of Closed Patterns-
It is complicated to derive a method to apply discovered patterns in text documents for information purifying systems. To make simpler this procedure, users first analyze the composition operation $\oplus$defined in [25]. Let p_1 and p_2 be sets of term-digit pairs. $p1 \oplus p2$ is called the composition of p1 and p2 which fulfills

$$p_1 \oplus p_2 = \{(c,d_1+d_2) | (c,d_1) \in p_1, (c,d_2) \in p_2\}$$

$$\cup \{(c,d) | (c,d) \in p_1 \cup p_2, not((c,\_) \in p_1 \cap p_2)\}$$

where is the wild card that matches any number. For the special case users have $p_1 \oplus = p$; and the operands of the composition operation are exchangeable. The result of the composition is quiet a set of term-number pairs.

For example,

$$\{(c_1,1),(c_2,2),(c_3,3)\} \oplus \{(c_2,4)\} = \{(c_1,1),(c_2,6),(c_3,3)\}$$

Or

$$\{(c_1,2\%),(c_2,5\%),(c_3,9\%)\} \oplus \{(c_1,1\%),(c_2,3\%)\} = \{(c_1,3\%),(c_2,8\%),(c_3,9\%)\}$$

Formally, for all positive documents $A_i \in D^+$, users first deploy its closed patterns on a common set of terms T in order to obtain the following d-patterns (deployed patterns, non-sequential weighted patterns):

$$\hat{A}_i = \{(c_{i1},n_{i1}),(c_{i2},n_{i2}),\ldots..,(c_{im},n_{im})\} \qquad (2)$$

Where c_{ij} in pair (c_{ij},n_{ij}) denotes a single term and $n_{ij}$is its support in A_i which is the total absolute supports given by closed patterns that contain c_{ij}; or n_{ij}(simply in this paper) is the total number of closed patterns that contain c_{ij} .

For example, using Fig. 1 and T1, users have

$sup_a(<c_3,c_4, c_6>)=3,$

$sup_a(<c_1,c_2>)= 3,$

$sup_a(<c_6>)=5,$and

$$\hat{A} = \{(c_1,3),(c_2,3),(c_3,3),(c_6,8)\}$$

The process of calculating d-patterns can be easily described by using the $\oplus$ operation in Algorithm 1 (PTM) displayed in Fig. 2, where a term's support is the total number of closed patterns that contain the term. Users also can obtain the d-patterns of the five sample documents in which are expressed as follows:

$\hat{A}_1$ = {(carbon,2),(emiss,1),(air,1),(pollut,1)},

$\hat{A}_2$ = {(greenhouse,1),(global,2),(emiss,1)},

$\hat{A}_3$ = {(greenhouse,1),(global,=1),(emiss,1)},

$\hat{A}_4$ = {(carbon,1),(airl,2),(antarct,1)},

$\hat{A}_5$ = {(emiss,1),(global,1),(pollut,1)}.

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

$$p(c) \longleftarrow p(c) \times \frac{1}{\sum_{c \in C} p(c)}$$

Actually the relationship between d-patterns and terms can be explicitly described as the following association mapping [25], a set-value function:

β:AAP$\rightarrow 2^{C\times[0,1]}$ (3)

such that

$\beta(p_i)$={(c_1, ω_1), (c_2, ω_2),……., (c_k, ω_k)}

for all $p_1 \in$ AAP, where

p_i={(c_1,f_1),(c_2,f_2),……,(c_k,f_k)} $\in$ AAP, $\omega_i = \frac{f_i}{\sum_{j=1}^{k} f_j}$

And C={c|(c,f) $\in$ p,p $\in$ AAP}

$\beta(p_i)$ is called the normal form (or normalized d-pattern)

of d-pattern p_1 in this paper, and

termset(p_i)={c_1,c_2,….,c_k}

4.4 Inner Pattern Evaluation

In this section, users [4] talk over how to restructuring supports of terms within normal forms of d-patterns centered on documents in the training set. The method will be useful to lessen the side effects of noisy patterns for the reason that of the low-frequency problem. This method is called inner pattern evolution here, for the reason that it only changes a pattern's term supports within the pattern.

A threshold is usually used to categorize documents into appropriate or inappropriate groups. Using the d-patterns, the threshold can be well-defined naturally as follows:

$$\text{Threshold(AAP)} = \min_{p \in AAP} \Big(\sum_{(c,\omega) \in \beta(p)} \text{support(c)} \Big) \quad (4)$$

A noise negative document A in D is a document that the system misleadingly recognized as a positive, that is weight (A)≥Threshold(AAP). In order to lessen the noise, users must track which d-patterns have been used to provide rise to such a mistake. Users call these patterns offenders of

nd.An offender of A is a d-pattern that has minimum one term in A. The set of offenders of A is defined by:

$$V(A)=\{p \in AAP | \textit{termset}(p) \cap A \neq \phi\}. \quad (5)$$

There are two sorts of offenders: 1) a complete conflict offender which is a subset of A; and 2) a partial conflict offender which holds part of terms of A.The basic idea of bring up-to-date patterns is described as follows:

Complete conflict offenders are detached from d-patterns first. For partial conflict offenders, their term supports are restructured in order to lessen the effects of noise documents. The main procedure of inner pattern evolution is executed by the algorithm IPEvolving. The input of this method is se of patterns .The output is a serene of d-pattern. Step 2 in IPEvolving is used to guess the threshold for discovery of the noise documents. Steps 3to 10 reread term supports by using all noise documents. Step 4 is to find noise documents and the equivalent offenders. Step 5 gets normal forms of d-patterns NDP. Step 6 calls algorithm scuffling to update NDP agreeing to noise documents. Steps 7 to 9 compose updated normal forms organized. The time complexity of Algorithm 2 is defined by step 2, the number of calls for Scuffling algorithm and the number of using $\oplus$ operation. Step 2 takes (nm).

For each noise pattern A, the algorithm catches its offenders that takes O(nm×|nd|)in step 4, and then calls once Scuffling. After that, it calls n $\oplus$ operation that takes

$$O(nmm)=O(nm)^2.$$

The task of algorithm Scuffling is to adjust the support supply of terms within a d-pattern. A different strategy is committed in this algorithm for each type of offender. As stated in step 2 in the algorithm Scuffling, complete conflict offenders (d-patterns) are detached since all elements within the d-patterns are held by the documents representing that they can be thrown away for preventing interfering from these possible "noises." The parameter proposing is used in step 4 for the purpose of provisionally keeping the cheap 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 act in a noise document. The algorithm calculates the base in step 5 which is definitely not zero since termset(p)−d≠ϕ; and then updates the support allocations of terms in step 6.

For example, for the following d-pattern

$\hat{A}$ ={(c_1,3),(c_2,3),(c_3,3),(c_3,4),(c_6,8)}.

Its normal form is

{(c_1,3/20),(c_2,3/20),(c_3,3/20),(c_4,3/20),(c_6,2/5)}

Assume nd={c_1,c_2,c_6,c_9}, $\hat{A}$ will be a partial conflict offender since

termset($\hat{A}$)∩nd=={t1,t2,t6}≠ϕ

Let μ =2,

offering= $\frac{1}{2}$ ×($\frac{3}{20}$ + $\frac{3}{20}$ + $\frac{2}{5}$)= $\frac{7}{20}$,and

base= $\frac{3}{20}$ + $\frac{3}{20}$ = $\frac{7}{10}$ Hence, users can get the

Following restructured normal form by using algorithm Scuffling: {(c_1,3/40),(c_2,3/40),(c_3,13/40),(c_4,13/40),(c_6,1/5)}

Let m = |T|, n=|D| the number of positive documents in a training set, and q be the number of noise documents in D. The time complexity of algorithm Scuffling is decided by steps 6 to 9. For a given noise document A, its time complexity is $O(nm^2)$if let A= A⋂ T, where T = {t∈termset(p)|p∈DP}. Therefore, the time complexity of algorithm Scuffling is $O(nm^2)$ for a given noise document. Based on the above study, the total time complexity of the inner pattern evolution is $O(nm+q(nm|nd|+nm^{2)=}O(qnm^2)$bearing in mind that the noise document A can be replaced by $A \cap T$ before leading the pattern evolution. The projected model contains two phases: the training phase and the testing phase. In the training phase, the suggested model first calls Algorithm PTM (D, min sup) to find d-patterns in documents (D) based on a *min sup*, and assesses term supports by deploying d-patterns to terms. It also calls Algorithm IPEvolving (D, DP, μ) to reread term supports using noise documents in D based on an trial coefficient μ. In the testing phase, it assesses weights for all entering documents using eq. (4). The entering documents then can be organized based on these weights.

5. ALGORITHMS USED

[11]**Algorithm 1**: SPMining(PL,min_sup)

Input: a list of nTerms frequent sequential pattern *PL*: minimum support *min_sup*.

Output: a set of sequential patterns *SP*.

Method:

1: $SP \leftarrow SP \leftarrow \{P_a \in | \exists P_b \in PL$ such a that $len(P_a) = len(P_b)-1$

$\wedge P_a \subset P_b \wedge supp_a(P_a) = supp_a(P_b)\}\}$ || *pattern mining*

2:$SP \leftarrow SP \cup PL$ ||*storing nTerms patterns*

3: $PL' \leftarrow \emptyset$

4: **for each** pattern *p* in *PL* **do begin**

5: generating p-projected database *PD*

6: **for each** frequent term *t* in *PD* **do begin**

7: $P' = p \bowtie t$ ||*sequence extension*

8: **if** $supp_r(P') \geq min\_sup$ **then**

9: $PL' \leftarrow PL' \cup P'$

10: **end if**

11: **end for**

12: **end for**

13: **if** $|PL'| = 0$ **then**

14: **return** ||*no more pattern*

15: **else**

16: **call** SPMining(*PL',min_sup)*

17: **end if**

18: **output** *SP*

Algorithm 2: PDM(*D*,min_sup)

Input: a list of document *D*: minimum support *min_sup*.

Output: a set of vectors Δ

METHOD:

1: $\Delta \leftarrow \emptyset$

2: **for each** document *d* in *D* **do begin**

3: extract *lTerms* frequent patterns *PL* from *d*

4: *SP* = SPMining(*PL,min_sup)* || Call Algorithm 1

5: $\vec{d} \leftarrow \emptyset$

6: **for each** pattern *p* in *SP* **do begin**

7: $\vec{d} \leftarrow \vec{d} \oplus P'$ || *P'* *is the expanded form of p*

8: **end for**

9: $\Delta \leftarrow \Delta \cup \{\vec{d}\}$

10: **end for**

Algorithm 3: PDS(*SP)*

Input: a set of frequent sequential patterns SP.

Output: a sevectors of feature in expanded form $\vec{d}$.

METHOD:

1: *sum_supp* = 0, $\vec{d} \leftarrow \emptyset$

2: **for each** pattern *p* in *SP* **do begin**

3: *sum_supp* += $sup_a(p)$

4: **end for**

5: **for each** pattern *p* in *SP* **do begin**

6: $f = supp_a(p)/(sum\_supp \times len(p))$

7: $P' \leftarrow \emptyset$

8: **for each** term *t* in *p* **do begin**

9: $P' \leftarrow P' \cup \{(t,f)\}$

10: **end for**

11: $\vec{d} \leftarrow \vec{d} \oplus P'$

12: **end for**

Algorithm 4: DPEvolving(Ω,*D*)

Input: a list of deployed patterns Ω; a list of documents *D*

Output: a set of term weight pairs $\vec{d}$.

METHOD:

1: $\vec{d} \leftarrow \phi$ // estimate minimum threshold

2: τ = *Threshold* (*D*)

3: **for each** document *d* in *D* **do begin**

4: **if** Threshold ({d})> τ then

5: $\Delta_p = \{Ap \in \Omega |$ termset (Ap)∩ d≠∅ }

6: Shuffling (d, Δ_p)

7: **end if**

8: **for each** deployed pattern d in Ω **do begin**

9: $\underline{\underline{d}} \leftarrow \underline{\underline{d}} \oplus Ap$

10**: end for**

11**: end for**

Algorithm 5: Shuffling (d, Δ_p)

Input: a document d and a list of deployed patterns Δ_p.

Output: updated deployed patterns.

METHOD:

1: **for each** deployed pattern *d* in Δ_p **do begin**

2: **if** termset(*d*) ⊆d then // complete conflict offender

3: Ω = Ω- {Ap}

4: else // partial conflict offender

5:*offering'* = $(1-\frac{1}{\mu})\times \sum_{t\in termset(Ap)} \{t.weight \mid t \in d\}$

6: base = $\sum_{t\in termset(Ap)} \{t.weight \mid t \notin d\}$

7: **for each** term *t* in termset(*Ap*) **do begin**

8: if *t* ϵ *d* then // shrink offender weight

9: *t*.weight = $\frac{1}{\mu}$ ×*t*.weight

10: **else** //shuffle weights

11: *t*.weight = *t*.weight×(1+*offering'*÷*base*)

12: **end if**

13: **end for**

14: **end if**

15: **end for**

6. EXPERIMENTAL RESULTS

T3: The List of Methods Used for Evaluation

Method	Description	Algorithm
Sequential ptns	Data mining method using sequential patterns	SPM
Sequential closed ptns	Data mining method using freq. sequential closed patterns	SCPM
Freq. Itemset	Data mining method using freq.itemset	NSPM
Freq.closed itemset	Data mining method using freq.closed itemset	NSCPM

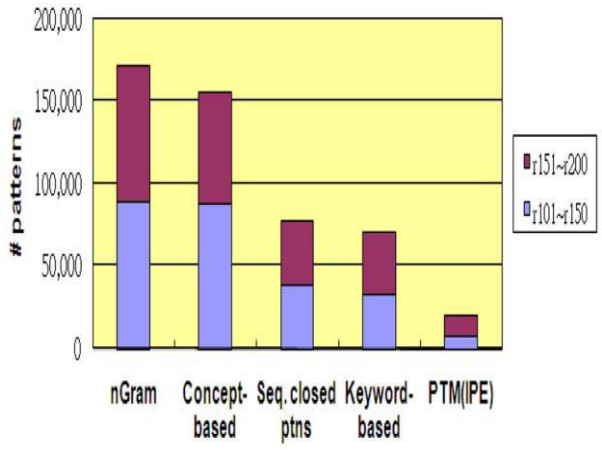


FIG: Comparison in the number of patterns used for training by each method on the first 50 topics (r101 r150) and the rest of the topics (r151 r200).

7. FUTURE SCOPE

Knowledge based system many useful features such as support and confidence of a pattern, relationship between patterns, distribution of pattern taxonomies, and the dimension of these taxonomies are provided. In PTM system, some features such as the relationship among patterns and support of patterns have been studied. The rest of the features will be used in further research work. Most of the data mining algorithms are computationally expensive such as PTM, especially during the phase of Pattern Deploying. One possible solution to improve the efficiency of pattern taxonomy-based model is to reduce the dimensionality of the feature space in the knowledge base. One alternative solution is to apply length-decreasing support constraints to frequent pattern mining

8. CONCLUSION

Many data mining techniques have been initiated in the last decade. These techniques carry (ASM) association rule mining, (CLOSET) closed frequent item set mining, maximum pattern mining, (SPM) sequential pattern mining and closed pattern mining. However, using these uncovered data (or patterns) in the field of text mining is hard to implement and not as much effective. This is because some useful long patterns with high specificity minimum support (i.e., the low-rate of occurrence problem). Users argue that not all recurrent short patterns are useful. Hence, misapprehension of patterns obtained from data mining techniques lead to the unsuccessful presentation. In this research work, an effectual pattern discovery technique has been established to overcome the low rate of occurrence and misapprehension problems for text mining. This proposed technique uses two processes, pattern evolving and pattern deploying, to refine the uncovered patterns in text documents. The exploratory results show that the proposed structure out performs not only other pure data mining-based process and the concept based structure, but also term-based state-of-the-art structures, such as BM25 and SVM-based structures

9. ACKNOWLEDGMENTS

The authors wish to thank Prof. Vibha Lahane from Dr. D Y Patil Institute of engineering and Technology, Ambi, Pune. Authors also wish to thank Ning Zhong, Yuefeng Li, and Sheng-Tang Wu for providing such a deep research on this topic.

10. REFERENCES

[1] Manan Parikh1, Bharat Chaudhari2 and Chetna Chand 2013 "A Comparative Study of Sequential Pattern Mining Algorithms"

[2] Jian Pei, Member, IEEE Computer Society, Jiawei Han, Senior Member, IEEE, BehzadMortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, UmeshwarDayal, Member, IEEE Computer Society, and Mei-Chun Hsu 2004 "Mining sequential Patterns by Pattern-Growth: The Prefix Span Approach"

[3] Jian Pei, Jiawei Han, and Runying Mao 2010 "CLOSET: An Efficient Algorithm for Mining Often Closed Item sets"

[4] RamshankarChoudhary Prof. AkhtarRasool Dr. NilayKhare2012 "Variation of Boyer-Moore String Matching Algorithm: A Comparative Analysis"

[5] li-ping jing,hou-kuan huang,hong-bo shi 2004 "improved feature selection approach tfidf intext mining"

[6] Robert Burbidge, Bernard Buxton2000's An Introduction to Support Vector Machines for DataMining

[7] Z. Yang,W. H. Tang,A. Shintemirov, and Q. H. Wu 2009 "Association Rule Mining-Based Dissolved Gas Analysis for Fault Diagnosis of Power Transformers"

[8] Gerard Salton and Christopher Buckley 1988 "Term-weighting Approaches in Automatic Text Retrival"

[9] Tsang-Hsiang Cheng and Chih-Ping Wei 2008 "A Clustering-Based Approach for Integrating Document-Category Hierarchies"

[10] RupaliBhaisare,T. RajuRao 2013 "Review On Text Mining With Pattern Discovery".

[11] NingZhong, Yuefeng Li, and Sheng-Tang Wu 2012 "Effective Pattern Discovery for Text Mining"

[12] JoydipDatta 2010 "Ranking in Information Retrieval"

[13] KjerastiAas Line Eikvil 1999 "Text categorization: a Survey"

[14] RamkrishnanShrikant, RakeshAgrawal "Mining Generelizedassotiation"

[15] Chih-Ping Wei and Yu-Hsiu Chang March 2007 "Discovering Event Evaluation Patterns From Document Sequences"

[16] Laura AuriaRouslanA.Moro August 2008 "Support Vector Machines (SVM) as a Technique for Solvency Analysis"

[17] http://en.wikipedia.org/wiki/Taxonomy_for_search_engines