ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 2, April 2017

Comprehensive Analysis of Large Text Document

B.Mahalakshmi, S.Saranya, S.Suganya, AL.Valliammai
K.S.Rangasamy College of Technology, Tiruchengode, Tamilnadu, India

Abstract—Many data mining techniques have been proposed for mining useful patterns in text documents. However, how to effectively use and update discovered patterns is still an open research issue, especially in the domain of text mining. Since most existing text mining methods adopted term-based approaches, they all suffer from the problems of polysemy and synonymy. Over the years, people have often held the hypothesis that pattern (or phrase)-based approaches should perform better than the term-based ones, but many experiments do not support this hypothesis. This paper presents an innovative and effective pattern discovery technique which includes the processes of pattern deploying and pattern evolving, to improve the effectiveness of using and updating discovered patterns for finding relevant and interesting information.

1 INTRODUCTION

Due to the rapid growth of digital data made available in recent years, 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 applications, 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.

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. In the beginning, 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 semantic meaning of many discovered terms is uncertain for answering what users want.

Over the years, people have often held the hypothesis that 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

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
Your ulti-MATE Research Paper !!!

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 2, April 2017

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.

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.

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, Prefix Span , FP-tree, SPADE, SLP Miner , 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 itemsets, 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.

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

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

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 sematic 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 betweenNonimportant terms and meaningful terms which describe a sentence meaning.

3 PATTERN TAXONOMY MODEL

We 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^+.

3.1Stem word, Stop word and Synonym word addition

Word and stem word is added in to 'stemword' table. Likewise, stop word is added into 'stopword' table. The word and its synonym word is added into synonym table.

3.2 Documents Collection

A set of text documents (D) are collected. Using the file open dialog control, the required text files are selected. Their names are stored in global variables which are used in further modules.

3.3 Paragraphs Extraction

The paragraphs are extracted from the documents D. They are referred as PS(d) where d is a single document. The Documents D contains positive D+ and negative D- documents in which the D+ documents contain the important term set which are of important.

3.4 Term Word Addition

The term words are added one by one to a list and are saved into the database.

3.5 Positive and Negative Documents Classification

From all the documents, the positive and negative documents are classified. The term values are keyed in and the documents containing those terms are treated as positive and all other documents are treated as negative documents.

3.6 Frequent and Closed Patterns

The frequent and closed patterns are identified. Given a term set X in document d,$\lceil x \rceil$ is used to denote the covering set of X for d, which includes all paragraph dp $\in$ PS(d) such that X$\subseteq$ dp. i.e.,$\lceil x \rceil = \{dp \mid dp \in PS(d), X \subseteq dp\}$.

Its absolute support is the number of occurrences of X inPS(d), that is $sup_a(X)=|\lceil x \rceil|$.Its relative support is the fraction of the paragraphs that contain the pattern, that is,

$$sup_r(X) = \frac{|\lceil x \rceil|}{|PS(d)|}.$$

A termset X is called frequent pattern if its sup_r (or sup_a) $\geq$ min sup, a minimum support.

TABLE 1: A SET OF PARAGRAPHS

PARAGRAPH	TERMS

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
Your ulti-MATE Research Paper !!!

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 2, April 2017

Dp1	**T1 T2**
Dp2	**T2 T4 T6**
Dp3	**T3 T4 T5 T6**
Dp4	**T3 T4 T5 T6**
Dp5	**T1 T2 T6 T7**
Dp6	**T1 T2 T6 T7**

Table 1 lists a set of paragraphs for a given document d, where PS(dp1, dp2, ... ,dp6}, and duplicate terms were removed. Let min_sup = 50%, we can obtain ten frequent patterns in Table 1 using the above definitions. Table 2 illustrates the ten frequent patterns and their covering sets.

TABLE 2: FREQUENT PATTERNS AND COVERING SETS

FREQUENT PATTERN	**COVERING SETS**
{T3 T4 T6}	**{DP2, DP3, DP4}**
{T3 T4}	**{DP2, DP3, DP4}**
{T3 T6}	**{DP2, DP3, DP4}**
{T4 T6}	**{DP2, DP3, DP4}**
{T3}	**{DP2, DP3, DP4}**
{T4}	**{DP2, DP3, DP4}**
{T1 T2}	**{DP1, DP5, DP6}**
{T1}	**{DP1, DP5, DP6}**
{TT2}	**{DP1, DP5, DP6}**
{T6}	**{DP2, DP3, DP4, DP5, DP6}**

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, it is believed that the shorter one, {t3;t4}, is a noise pattern and expect to keep the larger pattern, {t3;t4;t6}, only. Given a termset X, its covering set $\lceil x \rceil$is a subset of paragraphs. Similarly, given a set of paragraphs $Y \subseteq PS(d)$, it can be defined as its termset, which satisfies

Termset(Y)=$\{t \mid \forall\, dp \in Y \Rightarrow t \in dp\}$.

The closure of X is defined as follows:

Cls(X)= termset($\lceil x \rceil$).

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

Let X be a closed pattern. We can prove that$sup_a(X1) < sup_a(X)$, for all patterns $X1 \supset X$; otherwise, if $sup_a(X1) = sup_a(X)$, we have $\lceil x1 \rceil = \lceil x \rceil$.where $sup_a(X1)$ and $sup_a(X)$are the absolute support of pattern X1 and X, respectively. We also haveCls(X) = termset($\lceil x \rceil$) =termset($\lceil x1 \rceil$) $\supseteq X1 \supset X$that is, Cls(X)$\neq$ X.

3.7 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 = 50%. There are, however, only threeclosed patterns in this example. They are <t3; t4; t6>, <t1; t2>, and <t6>. Fig. 1 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 <t6> would become a direct sub pattern of <t3; t4; t6> after pruning nonclosed patterns.

Smaller patterns in the taxonomy, for example pattern <t6>, (see Fig. 1) 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.

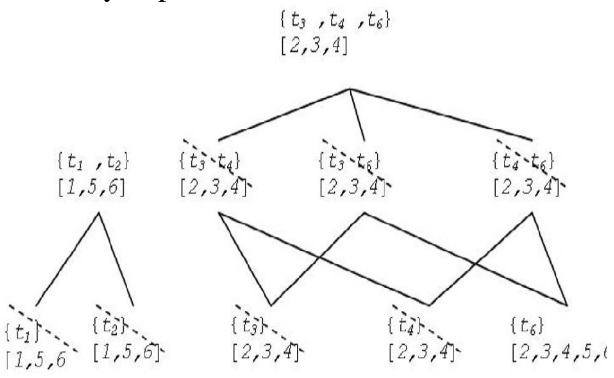


Fig. 1. Pattern taxonomy.

4. D-Pattern Mining Algorithm

To improve the efficiency of the pattern taxonomy mining, an algorithm, SPMining, was proposed in 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. 2 describes the training process of finding the set of d-patterns. For every positive document, the SPMining algorithm is first called in step 4 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. 2), all discoveredpatterns in a positive document are composed into a dpattern giving rise to a set of d-patterns DP in steps 6 to 9. Thereafter, from steps 12 to 19, term supports arecalculated based on the normal forms for all terms in dpatterns. Let m =lTl be the number of terms in T, n = $lD^{+}l$be the number of positive documents in a training set, K be the average number of discovered patterns in a positive document, and k be the average number of terms in a discovered pattern. We also assume that the basic operation is a comparison between two terms.

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

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

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)=∑ support(t)r(t,d),
t€T

wheresupport(t) is defined in Algorithm 1 (Fig. 2); and r(t,d) = 1 if t є d; otherwise r(t,d) = 0.

Algorithm 1: PTM (Dþ, min_sup).

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
Your ulti-MATE Research Paper !!!

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 2, April 2017

```
INPUT: Positive Documents $D^+$, minimum support (min_sup)
OUTPUT: d-patterns DP, and support of terms
1. DP = $\varnothing$.
2. foreach document d $\epsilon$ $D^+$ do
3.     let PS(d) be the set of paragraphs in d;
4.     SP = SPMining(PS(d),min_sup);
5.     d= $\varnothing$.
6.     foreach pattern $p_i \in$ SP do
7.         p = $\{ (t,1) \mid t \in p_i \}$;
8.         d = d $\oplus$ p;
9.     end
10.    DP = DP $\cup$ { d }.
11. end
12. T = $\{t \mid (t,f) \in p, p \in DP\}$;
13. foreach term $t \in T$ do
14.    support(t)=0;
15. end
16. foreach d-pattern $p \in DP$ do
17.    foreach $(t,w) \in \beta(p)$ do
18.        support(t)=support(t)+w;
19.    end
20. end
```

5. INNER PATTERN EVOLUTION

We discuss how to reshuffle supports of terms within normal forms of d-patterns based on negative documents in the training set. The technique will be useful to reduce the side effects of noisy patterns because of the low-frequency problem. This technique is called inner pattern evolution here, because it only changes a pattern's term supports within the pattern. A threshold is usually used to classify documents into relevant or irrelevant categories. Using the d-patterns, the threshold can be defined naturally as follows:

$$\text{Threshold}(DP) = \min_{p \epsilon DP} \Big(\sum_{(t,w) \epsilon \beta(p)} \text{support}(t) \Big)$$

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

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:

$$\Delta nd = \{p \ \epsilon \ DP \mid \text{termset}(p) \cap nd \neq \varnothing \}$$

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 arereshuffled in order to reduce the effects of noise documents.

The main process of inner pattern evolution is implemented by the algorithm IPEvolving (see Algorithm 2 in Fig. 3). The inputs of this algorithm are a set of d-patterns DP, a training set D = D^+U D^-. The output is a composed of d-pattern. Step 2 in IPEvolving 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 dpatternsNDP. Step 6 calls algorithm Shuffling (see Algorithm 3 in Fig. 4) to update NDP according to noise documents. Steps 7 to 9 compose updated normal forms together.

The time complexity of Algorithm 2 in Fig. 3 is decided by step 2, the number of calls for Shuffling algorithm and the number of using + operation. Step 2 takes O(nm). For each noise negative pattern nd, the algorithm gets its offenders that takes O(nm x |nd|)in step 4, and then calls once Shuffling. After that, it calls n+ operation that takes O(nmm) = $O(nm^2)$.

The task of algorithm Shuffling is to tune the support distribution of terms within a d-pattern. A different strategy is dedicated in this algorithm for each type of offender. As stated in step 2 in the algorithm Shuffling, complete conflict offenders (d-patterns) are removed since all elements within the d-patterns are held by the negative documents indicating that they can be discarded for preventing interference from these possible "noises."

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 ≠Ø; and then updates the support distributions of terms in step 6.

For example, for the following d-patternd = { (t1, 3), (t2,3), (t3,3), (t4,3), (t6,8)}.

Its normal form is

{(t1, 3/20),(t2, 3/20),(t3, 3/20),(t4, 3/20),(t6, 2/5)}.

Assume nd = (t1, t2, t6, t9), d will be a partial conflict offender since

Termset(d) ∩ nd = {t1,t2,t6}≠Ø.

Let μ= 2, offering =½ x (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:

{(t1, 3/40),(t2, 3/40),(t3, 13/40),(t4, 13/40),(t6, 1/5)}.

Let m = |T|, n = |D^+| the number of positive documents in a training set, and q be μthe 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 $O(nm^2)$if let nd =nd ∩ T, where T = {t ϵ termset(p)|p ϵ DP}. Hence, the time complexity of algorithm Shuffling is $O(nm^2)$for a given noise negative document.

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 ∩ 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 IPEvolving (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

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE) Vol. 3, Special Issue 2, April 2017

evaluates weights for all incoming documents using eq. (4). The incoming documents then canbe sorted based on these weights.

Algorithm 2: IPEvolving (Dþ, D_, DP, _).

INPUT: a training set $D = D^{+} \cup D^{-}$, a set of d-patterns DP, and an experimental coefficientμ

OUTPUT: a set of term-support pairs np.

```
1. np =$\varnothing$.
2. threshold = Threshold (D);
3. foreach noise negative document nd$\in D^{-}$ do
4.    if weight(nd)$\geq$ threshold then$\Delta$(nd), { p$\in$ DP| termset(p)$\cap$nd$\neq \varnothing$.
5.       NDP = {$\beta$(p)|p$\in$ DP};
6.  shuffling ( nd, $\Delta$(nd), NDP, $\mu$,NDP);//call Alg.3.
7. foreach  p$\in$ NDP do
8.       np=np$\oplus$ p;
9. end
10. end
```

Algorithm 3: Shuffling (nd, _ðndÞ, NDP, _. NDP).

Input: a noise document nd, its offenders Δ(nd), normal forms of D-pattern NDP, and an experimental coefficient μ

Output: Updated normal form of D-pattern NDP.

```
1. Foreach d-pattern p in $\Delta$(nd) do
2.       If termset(p)  C nd then NDP = NDP – {$\beta$(p)};//remove complete conflict offenders
3. Else//partial conflict offender
4. Offering = (1-1/$\mu$) x $\sum_{t \epsilon (termset(p) \cap nd)}$ support(t);
5. Base = $\sum_{t \epsilon (termset(p) - nd)}$ support(t);
6. Foreach term t in termset(p) do
7. If t $\epsilon$ nd then support(t) = (1/$\mu$) x support(t);//shrink
8. Else//grow supports
9. Support(t) = support(t) x (1 + offering ÷ base);
10. End
11. End
```

6 PERFORMANCE EVALUATION AND DISCUSSION

Weight of Document	DPM and INPM Algorithm (n)	EDPM and EINPM Algorithm (n)
200	155	176
250	220	234
300	272	291
350	322	342
400	383	398
450	429	445
500	468	489
550	523	547
600	578	592
650	633	646

In this study, Amazon Dataset collection is used to evaluate the proposed approach. Term stemming and stop word removal techniques are used in the prior stage of text

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 2, April 2017

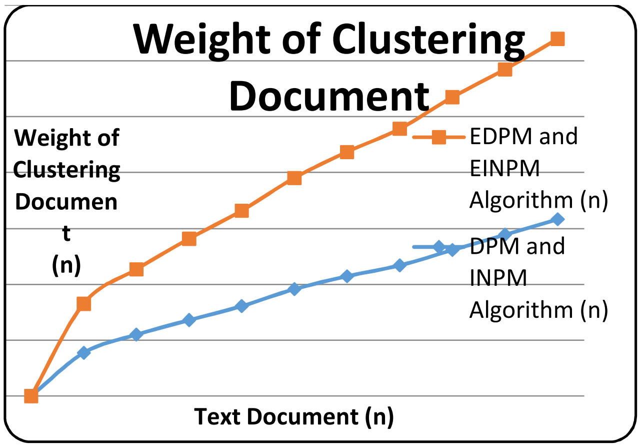
preprocessing. Several common measures are then applied for performance evaluation and our results are compared with the state-of-art approaches in data mining, concept-based, and term-based methods.

6.1 Experimental Data Set

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 is selected for suffix stripping. Terms with term frequency equaling to one are discarded.

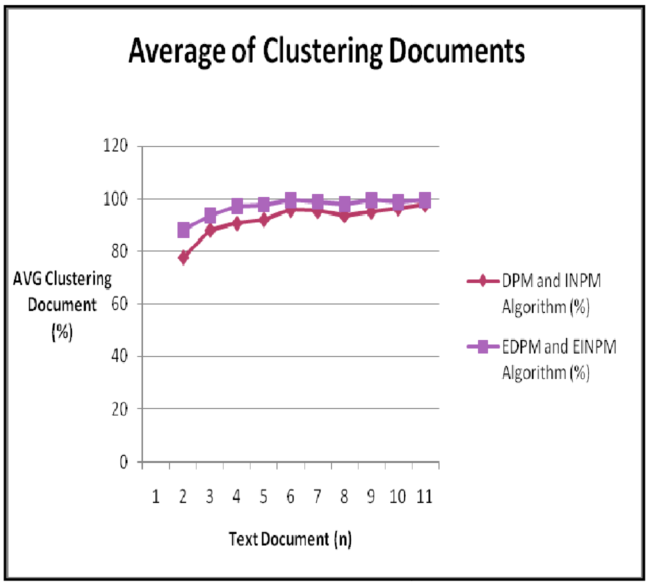
6.2 Experimental Results



Weight of Document	DPM and INPM Algorithm (n)	EDPM and EINPM Algorithm (n)
200	155	176
250	220	234
300	272	291
350	322	342
400	383	398
450	429	445
500	468	489
550	523	547
600	578	592
650	633	646

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 2, April 2017



7 CONCLUSIONS

Many data mining techniques have been proposed in the last decade. These techniques include association rule mining, frequent itemset 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.

REFERENCES

[1] W. Lam, M.E. Ruiz, and P. Srinivasan, “Automatic Text Categorization and ItsApplication to Text Retrieval,” IEEE Trans. Knowledge and Data Eng., vol. 11, no. 6, pp.865-879, Nov./Dec. 1999.

[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] Y. Li and N. Zhong, “Mining Ontology for Automatically Acquiring Web User Information Needs,” IEEE Trans. Knowledge and Data Eng., vol. 18, no. 4, pp. 554-568,Apr. 2006.

[4] Angeles Maria Del Pilar et al, “Solving Date inconsistencies and Data Integretionwith the Data Quality Manager” conference on Data Mining,Vol.17,pp.464-478,2010.

[5] Anindya Ghose and Panagiotis Ipeirotis. G,” Estimating the Helpfulness andEconomic Impact of Product Reviews: Mining Text and Reviewer Characteristics” IEEE Transactions on Knowledge and Data Engineering,Vol. 23, No. 10,pp.377-387 October 2011.

[6] Dursun Delen , Asil Oztekin “Introduction to Data, Text, and Web Mining forManagerial Decision Support Mini-track” Hawaii International Conference on System Science Vol.8,pp.198-207, 2014.

[7] Heimerl. F,Koch. S,Bosch. H,and Ertl. T "Visual Classifier Training for TextDocument Retrieval". IEEE Transactionson Visualization and Computer Graphics,Vol.18(12),pp:2839–2848,December 2012.

[8] Hu. X ,Bradel. L,Maiti, L.House. D and North. C "Semantics of directly manipulating spatializations" .Visualization and Computer Graphics, IEEETransactionson,Vol.19(12)pp:2052–2059,2013.

[9] Kang Liu, Liheng Xu, and Jun Zhao "Co-Extracting Opinion Targets and Opinion
Words from Online Reviews Based on the Word Alignment Model" IEEETransactions on Knowledge and Data Engineering, Vol. 27,No. 3,pp.200-210, March 2015.

[10] Kenric Hammond. W, Ryan Laundry. J "Application of a Hybrid Text MiningApproach to the Study of Suicidal Behavior in a Large Population" 11th ACM SIGKDD Int'l Conference Knowledge Discovery in Data Mining,Vol.10, pp.198-207, 2013.

[11] Keimand. D." A new method for visual literary analysis". IEEE Symposium on Visual Analytics Science and Technology, Vol.14, pp:115–122, October 2007.

[12] Kong. X "Context-Based Collaborative Filtering for Citation Recommendation"IEEE Transaction on Knowledge and Data Engineering, vol.17, no.6, pp.734-749,Jun. 2015.

[13] Lexing Xie, Apostol Natsev, Xuming He, John Kender. R, Matthew Hill, andJohn Smith. R" Tracking Large-Scale Video Remix in Real-World Events" IEEE Transactions on Multimedia, Vol. 15, No. 6,pp.234-241 October 2013.

[14] Liu. S, Zhou. M. X, Pan. S, Qian. W, Cai. W, and Lian. W. "Interactive, topicbased visual text summarization and analysis". ACM Conference on Information and Knowledge Management,Vol.14, pp.543–552,2009.

[15] Manabu Torii, Cecilia Arighi. N, Gang Li, Qinghua Wang, Cathy Wu. H, and Vijay-Shanker. K "RLIMS-P 2.0: A Generalizable Rule-Based InformationExtraction System for Literature Mining of Protein Phosphorylation Information"IEEE/ACM Transactions on Computational Biology and Bioinformatics,Vol. 12,No. 1,pp.642-664 January 2015.

[16] Massa. P and Avesani. P, "A Text Mining Approach to Automated Healthcarefor the Masses" Knowledge-Based System,Vol.23,No.3,pp.232-238,2010.

[17] Manabu Torii, Cecilia Arighi. N, Gang Li, Qinghua Wang, Cathy Wu. H, and Vijay-Shanker. K "RLIMS-P 2.0: A Generalizable Rule-Based InformationExtraction System for Literature Mining of Protein Phosphorylation Information"IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 12, No. 1,pp.1251-1257, January 2015.

[18] Morinage. S and Yamanishi. K ,"Tracking Dynamics of Topic Trends using

a finite mixture model" ,10th ACM SIGKDD International Conference onKnowledge Discovery and Data Mining,Vol.12, pp.811-816,2013.

[19] Ning Zhong, Yuefeng Li, and Sheng-Tang Wu "Effective Pattern Discoveryfor Text Mining" IEEE Transactions on Knowledge and Data Engineering, Vol. 24, No. 1,pp.464-478 January 2012.

[20] Riccardo Scandariato, James Walden, Aram Hovsepyan and Wouter Joosen "Predicting Vulnerable Software Components via Text Mining"ACM Transactions on Knowledge Discovery Data, Vol.4, No.1,pp.1:1-1:24,2010.