ISSN:0975-9646

Effective Pattern Identification Approach for Text Mining

Vaishali Pansare

Computer Science and Engineering,

Jawaharlal Nehru Engineering College, Aurangabad-431003, M. S., India

***Abstract* — Text mining is an important research area of data mining. Many data mining techniques have been discovered for finding useful patterns in text document. These techniques include association rule mining, frequent item set mining, sequential pattern mining, maximum pattern mining and closed pattern mining. It is challenging task to find and extract user required information in text documents effectively. Different text mining methods are text based method, phrase based method, concept based method and pattern based method. These methods are based on how the text document is analyzed. This paper addresses the use of association rule mining based AprioriAll algorithm for discovering frequent patterns in text documents effectively. In proposed system, a pattern based approach is used because discovered patterns are more specific than whole documents. The proposed system is time efficient.**

Keywords — Text Mining, Pattern Identification, HashTree, WorldNet 2.1, d-pattern Mining

I. INTRODUCTION

Data mining is defined as a non-trivial extraction of implicit, previously unknown and potentially useful information from data. The purpose of data mining is to extract interesting knowledge from the large database. Nowadays most of the information in business, government, industry and other institutions is stored in text form into database and this text database contains semi structured data. Text mining is a variation on a field called data mining that tries to find interesting patterns from large databases. In traditional search the users typically look for already known terms which have been written by someone else. In this search, the problem occurs in result as it is not relevant to users need. This is the goal of text mining to discover unknown, useful and relevant information which is not known and yet not written down. Many text mining methods have been developed for retrieving useful information. These methods include term based method, phrase based method, concept based method and pattern based method [4]. Term based method suffers from the problems of polysemy and synonymy. Phrase-based approaches perform better than the term based because more information is carried by a phrase than by a term. Still there are some disadvantages of phrase-based approach such as occurrence of phrases, noisy phrases among them and low frequency of occurrence. To overcome disadvantages of phrase-based approaches, pattern mining-based approaches have been proposed. Pattern discovery is used as an effective technique for knowledge discovery in many applications. In this paper, we focus on the development of an efficient mining algorithm to effectively extract and use the discovered patterns and apply it to the field of text mining. The pattern based method is used in proposed system.

II. RELATED WORK

A. *Literature Review*

Reference [1] shows an innovative and effective pattern discovery technique which includes processes of pattern deploying and pattern evolving, to improve the effectiveness of using and updating discovered patterns for finding relevant and interesting information. In [2], the main focus is on developing an efficient mining algorithm, pattern discovery technique which includes processes of pattern deploying and pattern evolving. Reference [3] shows an innovative technique, pattern taxonomy mining to improve the effectiveness of using discovered patterns for finding useful information. A proposed system in [4] increases efficiency of pattern discovery using different data mining algorithms with pattern deploying and pattern evolving method to solve misinterpretation and low frequency problem. Performance enhancement is achieved by applying multithreaded approach and concept formation from effective discovered patterns. In [5], taxonomy of sequential pattern mining techniques has presented in the literature with web usage mining as an application. This article investigates these algorithms by introducing taxonomy for classifying sequential pattern mining algorithms based on important key features supported by techniques. Reference [6] shows an efficient mining algorithm like pattern taxonomy model to find particular pattern within a reasonable and acceptable time frame. Reference [7] shows a general approach to generate such an annotation for a frequent pattern by constructing its context model, selecting informative context indicators, and extracting representative transactions and semantically similar patterns. In [8], specific documents are preprocessed before placing patterns discovery and preprocessing the document dataset using tokenization, stemming and probability filtering approaches. In [9], a method that use WordNet concept to categorize text documents is proposed. This proposed method extracts generic concepts from WordNet for all the terms in the text then combines them with terms in different ways to form a new representative vector. Reference [10] shows the information retrieval approach using pattern based method which uses pattern deploying and pattern evolving techniques. Reference [11] shows three variations of Apriori algorithm using data structures hash tree, trie and hash table trie i.e. trie with hash technique on MapReduce paradigm. Reference[12]

shows us how one can accurately discover and release the most significant patterns along with their frequencies in a data set containing sensitive information, while providing rigorous guarantees of privacy for the individuals whose information is stored there. Reference [13] analyze the pattern using different algorithms like Apriori, Hash tree and Fuzzy and then we used enhanced Apriori algorithm to give the solution for Crisp Boundary problem with higher optimized efficiency while comparing to other algorithms.

B. Text Mining Methods

Text mining methods are developed on the basis of how text document is analyzed. These methods are as follows:

1) Term based method
2) Phrase based method
3) Concept based method
4) Pattern based method

1) Term based method: The term in document is used to identify the content of text. Each term is associated with the value known as weight. In term based method text document is analyzed on the basis of the term. The advantages of term based method include efficient computational performance as well as mature theories for term weighting. However, term based method suffers from the problems of polysemy and synonymy, where polysemy means a word has multiple meanings, and synonymy is multiple words having the same meaning.

2) Phrase based method: The phrased-based approaches could perform better than the term based ones, as phrases may carry more "semantics" like information. This hypothesis has not fared too well in the history of IR. Although phrases are less ambiguous and more discriminative than individual terms, there are some reasons for the discouraging performance of this method. The reasons are as follows:

1) There are large numbers of occurrence of phrase and noisy phrases among them.
2) Phrases have low frequency of occurrence.
3) Phrases have inferior statistical properties to terms.

To overcome the disadvantages of phrase-based approaches, pattern mining-based approaches have been proposed, which adopted the concept of closed sequential patterns and pruned non closed patterns.

3) Concept based method: It is important to find term that contributes more semantic meaning to document. This concept is known as concept based method. In concept based method the term which contributes to sentence semantic is analyzed with respect to its importance at sentence and document levels.

4) Pattern based method: In this method the text documents are analysed on the basis of pattern. Patterns are item subsequences, sets, or substructures that appear in a data set with frequency no less than a user specified threshold. The pattern used as a word or phrase that is extracted from the text document. To overcome the disadvantages of phrased-based approaches, pattern mining based approaches have been proposed.

C. Terms and Concepts

1) Text Mining

Text mining is defined as the process of extracting interesting, useful and non-trivial patterns or knowledge from text documents. Text mining is also known as text data mining or knowledge discovery from textual databases. Text mining is nothing but data mining, as the application of algorithm as well as methods from the machine learning and statistics to text with goal of finding useful pattern. Text mining methods include term based method, phrase based method, concept based method and pattern based method. Knowledge discovery can be effectively use and update discovered patterns and apply it to field of text mining.

2) Data Mining

Data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. Different data mining techniques have been proposed for extracting text data. These techniques include association rule mining, frequent item set mining, sequential pattern mining, maximum pattern mining and closed pattern mining. Data mining is the principle of sorting through large amounts of data and picking out required information. It has been described as finding hidden information in a database. Data mining is usually used by business intelligence organizations, and financial analysts, but it is increasingly used in the sciences to extract information from the enormous datasets generated by modern experimental and observational methods.

3) Pattern Discovery

Pattern discovery in text mining is an important research task in the field of data mining. There are large numbers of pattern which may be discovered from a text document, but not all of them are interesting. A pattern is called knowledge if it is interesting and certain enough, according to the user's imposed interestingness measures and criteria. Frequent patterns are itemsets, subsequences, or substructures that appear in a data set with frequency no less than a user-specified threshold. Discovered patterns are useful and interesting knowledge is extracted from large amount of data. A system may encounter a problem where a discovered pattern is not interesting to a user. Such patterns are not qualified as knowledge. Therefore a knowledge discovery system should have the capability of deciding whether a pattern is interesting enough to form knowledge in the current context [2].

4) WordNet 2.1

WordNet is ontology of cross-lexical references whose design was inspired by the current theories of human linguistic memory. The difference between WordNet and the traditional dictionary is the separation of the data into four databases associated with the categories of verbs, nouns, adjectives and adverbs. Each database is differently organized than the others. The names are organized in hierarchy, the verbs by relations, the adjectives and the adverbs by N-dimension hyperspaces [9]. English names, verbs, adjectives and adverbs are organized in sets of synonyms (synsets), representing the underlying lexical concepts. The synsets are sets of synonyms which gather

lexical items having similar significances. Sets of synonyms are connected by relations. WordNet covers most names, verbs, adjectives and adverbs of the English language. The latest version of WordNet (2.1) is a vast network of 155000 words, organized in 117597 synsets. There is a rich set of 391.885 relations between the words and the synsets and between the synsets themselves. The basic semantic relation between the words in WordNet is synonymy. WordNet is used in many text classification methods as well as in Information Retrieval (IR) because of its broad scale and free availability. Studies in which the synsets of WordNet were used as index terms have very promising results.

5) HashTree Data Structure

HashTree is rooted (downward), directed tree. Hash tree contains two types of nodes, inner nodes and leaves. Leaves of hash tree contain a list which stores candidates [11]. HashTree requires no initial root hash table yet are faster and use significantly less space than chained or double hash trees. A Hash tree stores all candidate k-item sets and their counts. The root is empty and its children are the frequent 1-itemsets. Any node at depth = k will denote a frequent k-itemset. An internal node v at level m contains, bucket pointers. This tells that which branch is the next one to be traversed. The hash of the m'th item is used to decide this.

Join step using HashTree:

Only the frequent k-1 item sets, which have common parents, should be considered for the joining step. So checking all k-1 item sets in Lk-1 is avoided.

Prune step using HashTree:

To determine if a k-1 itemset is frequent, we have to look only for those item sets that have common parents, and thus avoid going through all k-1 item sets in Lk-1 [13]. To overcome crisp boundary problem, find out the min support, Scan D and count each itemset in Ck, if the count is greater than min support, then add that itemset to Lk. Joining and pruning of itemsets and checking subset of each transaction against candidates are very computation intensive process in Apriori algorithm. Also a large number of candidates require large memory during execution of algorithm. HashTree is the central data structure which reduces the computation cost as well as organizes candidate itemsets in a compact way in memory. Therefore, it is an efficient data structure.

7) d-pattern Mining

This technique is an important technique used in pattern based text mining. A d-pattern mining technique is used to summarize the discovered patterns. It evaluates specificities of patterns and then evaluates term-weights according to the distribution of terms in discovered patterns.

III. IMPLEMENTATION

A. Proposed System Design

The various modules of Effective pattern identification in text mining are as follows:

1. Loading documents

In this module, we load text documents as per our need. Then user can retrieve any one of the document. This document is given to next process which is nothing but preprocessing.

2. Text Preprocessing

The retrieved text document preprocessing is done in this module with help of two sub processes such as:

a) Stop words removal

b) Text stemming

Stop words are words which are filtered out prior to, or after, processing of natural language data. Stemming is the process for reducing inflected (or sometimes derived) words to their stem base or root form. It is generally a written word forms.

3. Apply AprioriAll

AprioriAll is a sequential data pattern discovery algorithm. In this module the text documents are split into paragraphs and each statement is considered to a single document paragraph which leads to the set of terms which can be extracted from set of text documents. It involves a sequence of five phases that work together to uncover sequential data patterns in large datasets. The five phases are sort phase, L-itemset, transformation, sequence, Maximal phase.

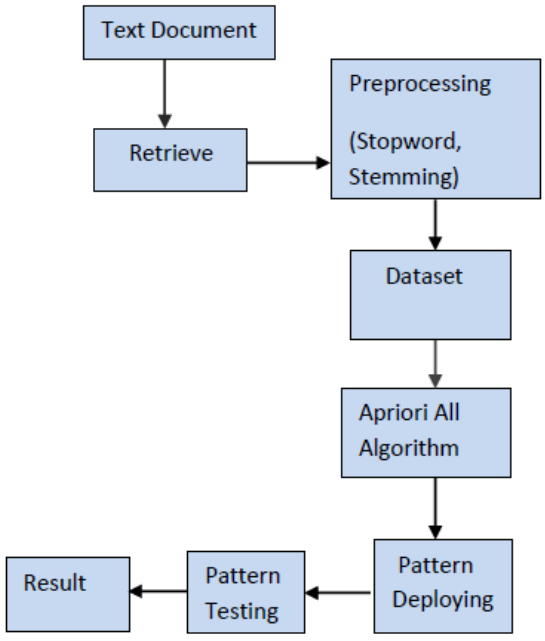
4. Pattern Deploying

The technique such as d-pattern mining technique is used to summarize the discovered patterns. It evaluates specificities of patterns and then evaluates term-weights according to the distribution of terms in discovered patterns. It solves Misinterpretation Problem. Term support means weight of the term is evaluated. Term supports are calculated by all terms in d-pattern which is used to discover all patterns in positive documents are composed.

5. Pattern Testing

This module is used to identify the noisy patterns in documents. Sometimes, system falsely identified negative document as a positive. So, noise is occurred in positive document. The noised pattern named as offender. In positive documents, the reshuffle process is done in case of partial conflict offender.

B. Proposed System Block Diagram



C. Working

Association rule mining based AprioriAll algorithm is used in proposed system. It uses principle of Apriori algorithm. According to this AprioriAll algorithm, in each pass, we use the large sequences from the previous pass to generate the candidate sequences and then measure their support by making a pass over the database. At the end of the pass, the support of the candidates is used to determine the large sequences. In the first pass, the output of the l-itemset phase is used to initialize the set of large l-sequences. The candidates are stored in HashTree to quickly find all candidates contained in a document sequence.

Let a database of text documents in the dataset. After preprocessing step, the processed text document is taken as input to AprioriAll algorithm. The extracting patterns from text documents process using this algorithm was decomposed with five steps:

a) Sort step: This step sort the textual database according to document paragraph (dp) and their corresponding itemsets. Each single textual statement is considered to be a single document paragraph in text document.

Where d= document; dp = set of document paragraph, t= term;

TABLE I A SET OF DOCUMENT PARAGRAPH

Document Paragraph(dp)	Data Items in Document Paragraph
dp1	t1,t3,t5,t6
dp2	t5,t6,t2
dp3	t3,t5,t7,t8
dp4	t6,t2,t3,t4
dp5	t5,t7,t3
dp6	t4,t1,t3,t5
dp7	t3,t1,t4

TABLE II FREQUENT PATTERN AND COVERING SET

Frequent Pattern	Covering Set
t5,t7,t3	dp3,dp5
t1,t3,t5	dp1,dp6
t3,t1	dp6,dp7
t3,t4	dp4,dp7
t3,t5	dp1,dp3,dp5
t5,t6	dp1,dp2
t6,t2	dp2,dp4

b) L-itemset step: In this step, the objective is to obtain the large 1-itemsets from the sorted textual database, based on the minimum support threshold.

c) Transformation step: This step replaces the sequences by those large itemsets they contain. For efficient mining, all the large itemsets are mapped into an integer series. Finally, the original database will be transformed into set of 1-itemset sequences represented by those large itemsets.

d) Sequence step: From the transformed sequential database, this step generates all frequent sequential patterns using Apriori-like algorithm.

Apriori-like algorithm:

Ck: Candidate itemset of size k

Lk: frequent itemset of size k

L1 = frequent items

1. L1 = {frequent 1-sequences}; // result of the itemset phase

2. for (k=2; Lk-1≠ □; k++) do

3. begin

4. Ck = candidates generated from Lk-1 using Apriori-generate function

For each large-sequence c in the database do increment the count of all candidates in Ck that are contained in c

6. Lk = candidates in Ck with minimum support

7. End.

Answer = maximal sequences in Uk Lk;

Apriori-generate function:

1. Insert into Ck

2. Select p.litemset1, ..., p.litemsetk-1, q.litemsetk-1

from Lk-1 p, Lk-1 q

where p.litemset1 = q. litemset1, ...,

p.litemsetk-2 = q.litemsetk-2;

Apriori-generate function

1. Join step: Join Lk-1 with Lk-1 with the join condition that the first k-2 itemsets should be the same. No more lexicographic ordering is preserved and allows a sequence to be joined with itself.

2. Prune step: Delete all candidates that have non frequent subsequences.

e) Maximal step: This step prunes the sequential patterns that are contained in other super sequential patterns, because we are only concerned with maximum.

An interesting downward closure property, named Apriori, among frequent k-itemsets: A k-itemset is frequent only if all of its sub-itemsets are frequent. This property means that frequent itemsets can be mined by identifying frequent 1-itemsets (first scan of the database), then the frequent 1-itemsets would be used to generate candidate frequent 2-itemsets and so on.

IV. RESULT AND DISCUSSION

A. Performance Analysis

In this paper, we have checked the performance of the proposed system by using two parameters- time and minimum support. Fig.3 shows the execution time required for three methods PTM, Apriori and AprioriAll using three input text files with different sizes. Here the minimum support is same for three algorithms. We can see that AprioriAll requires less time compared to other algorithms. Hence the proposed AprioriAll algorithm is time efficient.

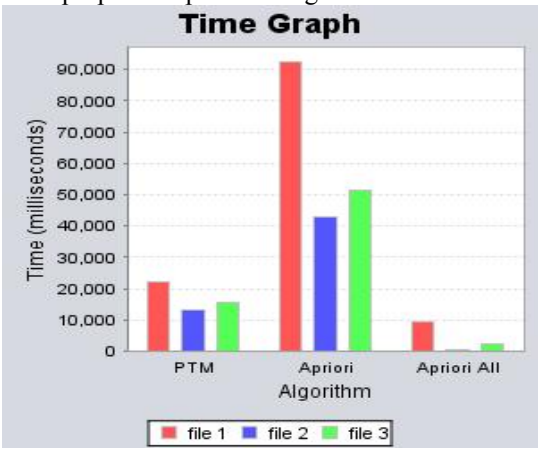


Fig.1: Time Comparison with Algorithm for same minimum support

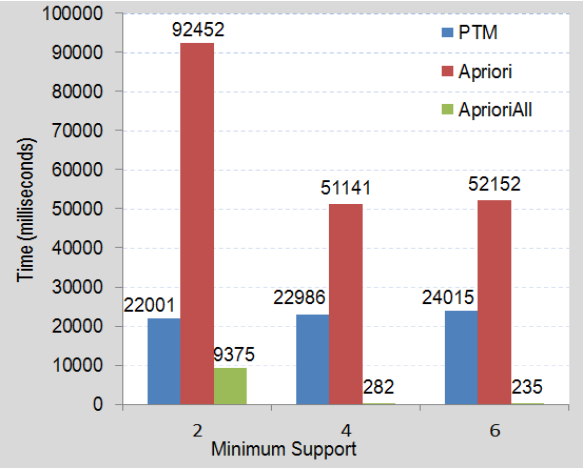


Fig. 2: Time Comparison with Minimum Support

We have observed the execution time for three methods on a single input text file for different minimum support values. Fig.4 shows the execution time required for three methods PTM, Apriori and AprioriAll for different minimum support values 2, 4 and 6. In this case, AprioriAll takes less time than other methods. As minimum support increases, the time of AprioriAll decreases for given input text file. This shows that AprioriAll becomes more time efficient than PTM and Apriori algorithm for different minimum support values.

V. Conclusion

In this research work, we have mainly focused on developing efficient mining algorithm for discovering patterns in text documents and search for useful and interesting patterns. In proposed technique we can take different formats of text file as input. AprioriAll algorithm is used in proposed system. It uses HashTree structure to store candidate itemsets. It helps to find frequent patterns and itemsets within less time. Patterns are more specific and carry more information than terms. So pattern mining based technique is used in proposed system. This system solves low frequency and misinterpretation problem. The results have shown that the execution time required by AprioriAll algorithm is less than time required by other two algorithms. Hence AprioriAll is more time efficient compare to PTM and Apriori algorithm. This method can be further modified to take into account large input text files.

References

[1] Ning Zhong, Yuefeng Li, and Sheng-Tang Wu, "Effective Pattern Discovery for Text Mining", *IEEE Transactions on Knowledge And Data Engineering*, Vol. 24, No.1, January 2012.

[2] Dipti Charjan and Mukesh Pund, "Pattern Discovery For Text Mining Using Pattern Taxonomy", *International Journal of Engineering Trends and Technology (IJETT)*, Volume 4, Issue 10, October 2013.

[3] Yuefeng Li, Sheng-Tang Wu and Xiaohui Tao, "Effective Pattern Taxonomy Mining in Text Documents", *ACM*, October 26–30, 2008.

[4] Sonali Gaikwad and Archana Chaugule, "Performance Enhancement for Effective Pattern Discovery & Concept Formation in Text Mining", *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 4, Issue 9, September 2014.

[5] Nizar Mabroukeh and C. I. Ezeife, "A Taxonomy of Sequential Pattern Mining Algorithms", *ACM Computing Surveys*, Vol. 43, No. 1, Article 3, November 2010.

[6] A. D. Khade, A. B. Karch, D. S. Jadhav and A. S. Zore, "Discover Effective Pattern for Text Mining", *International Journal of Engineering Research and Applications*, Vol. 4, Issue 3 (Version 2), March 2014.

[7] Qiaozhu mei, Dong Xin, Hong Cheng, Jiawei Han and Chengxiang Zhai, "Semantic Annotation of Frequent Patterns", *ACM Transactions on Knowledge Discovery from Data*, Vol. 1, No. 3, December 2007.

[8] Pattan Kalesha, M. Babu Rao and Ch. Kavitha, "Efficient Preprocessing and Patterns Identification Approach for Text Mining", *International Journal of Computer Trends and Technology (IJCTT)*, Vol. 6, No. 2, December 2013.

[9] Zakaria Elberrichi, Abdelattif Rahmoun and Mohamed Bentaalah, "Using WordNet for Text Categorization", *International Arab Journal of Information Technology*, Vol. 5, No. 1, January 2008.

[10] Vishakha Bhope and Sachin Deshmukh, "Information Retrieval using Pattern Deploying and Pattern Evolving Method for Text Mining", *International Journal of Computer Science and Information Technologies (IJCSIT)*, Vol. 6 (4), 2015.

[11] Sudhakar Singh, Rakhi Garg, P.K. Mishra, "Performance Analysis of Apriori Algorithm with Different Data Structures on Hadoop Cluster", *International Journal of Computer Applications*, Vol. 128, No. 9, October 2015.

[12] Raghav Bhaskar, Srivatsan Laxman and Adam Smith, "Discovering frequent patterns in sensitive data", *ACM*, July 2010.

[13] S.Veeramalai, N.Jaisankar and A.Kannan "Efficient Web Log Mining Using Enhanced Apriori Algorithm with Hash Tree and Fuzzy" *International journal of computer science & information Technology (IJCSIT)*, Vol.2, No.4, August 2010.

[14] Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques", Second Edition, *University of Illinois at Urbana-Champaign.*