THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

Pattern Discovery for Text Mining

S. S. Patil
Department of Computer
Bharati Vidyapeeth University College of Engineering, Pune, India
V. M. Gaikwad
Associate Professor, Department of Computer
Bharati Vidyapeeth University College of Engineering, Pune, India

Abstract:
In this paper data mining skills are available which are useful in mining the sample of text document. However, competent studies of a new pattern which is helpful and correct the discovered patterns are still an open research topic, particularly in the text mining field. Right now text mining methods have term based methods and they have difficulty of polysemy and synonymy. To decrease these difficulties new method build is phrase based method. The phrase based method is better than the term based, but many experiments are not done in a phrase based method.
To present the new and efficient pattern discovery system, this has a procedure of pattern deploying and evolving. It improves the effectiveness and updating new pattern to discover suitable and related information.

1. Introduction

Knowledge discovery is method of nontrivial extraction of knowledge from huge database. In the existing years, there is fast growth in digital data, thus it's needful to show these information into useful information and data. By the employment of this discovery the information and data extracted from large quantity of information could be advantageous for market research and business management. Thus data mining is an important step in the process of information discovery in large database.
In a last five to ten years, to perform completely different knowledge task the following data mining techniques are applied.

- Association rule mining,
- Frequent item set mining,
- Sequential pattern mining,
- Maximum pattern mining, and
- Closed pattern mining.

All these techniques focus on efficient mining algorithms used to find particular pattern that are inexpensive and acceptable in timeframe. Using these techniques, a variety of patterns of data mining are generated. How these techniques effectively use is an open research issue. However, there is now scope for the research. We are focusing on the development of a knowledge discovery model which effectively use and update the patterns. These patterns are discovered and apply to text mining.
In the text mining it is a difficult for user to find exact information (or features) from text documents. Firstly, information Retrieval (IR) provided number of term-based ways to solve this problem, such as Rocchio Classifier and probabilistic models, rough set models, BM25 and support vector machine (SVM) based filtering models. The advantages of term- based methods include clever computational performance as well as mature theories for term weighting. They have emerged from the last twenty years from the IR and communities of machine learning. However, term- based ways suffer from the issues of synonymy and polysemy, wherever synonymy is multiple words having identical meanings and polysemy means a word has multiple meanings. The meaning of the many discovered terms is unsure about what user wants. Over the years, individuals have usually control the hypothesis that phrase-based approaches might perform best than the term-based ones, as phrases could carry a lot of "semantics" like information. This hypothesis has not fared too well within the history of IR. Though phrases are less ambiguous and a lot of discriminating than individual terms, the explanation for the discouraging performance as follows:

- Phrases have low applied mathematics properties in terms,
- They need frequent of occasion, and
- There are more numbers of excessive and hissing phrases among them.

In the company of those setbacks, sequential patterns utilized in the data mining community have turned out to be a promising another choice to phrases. As a result of sequential patterns get pleasure from applied mathematics properties like terms. Pattern mining-based approaches are used to beat the disadvantage of phrase-based approach. Pattern mining-based approaches adopted the idea of the closed sequential patterns and also the non closed patterns. These pattern mining primarily based approaches had shown some extent enhancements on the effectiveness.

There are two basic things concerning the effectiveness of pattern-based approach low frequency and mistaking. Given a special topic, an extremely frequent pattern is sometimes a general pattern, or a particular pattern of low frequency. If we don't offer the support, lots of noisy patterns would be searched. Mistaking means that the measures utilized in pattern mining (e.g., "support" and "confidence") that is not appropriate for discovering patterns to provide answer, what we would like. The major problem for this is how to use discovered patterns to exactly evaluate the weights of useful information in text documents.

Over the years, IR has developed several mature techniques that demonstrate the terms which are essential features in the text documents. However, general terms have larger weights (e.g., the term frequency and inverse document frequency (tf*idf) weighting scheme) because they can be often used in both related and unrelated information. For example, term "JDK" may have smaller weight than "LIB" in a certain data collection, but we believe that term "LIB" is less specific than term "JDK" for describing "Java Programming Language". Therefore, it is not sufficient for evaluating the weights of terms on their classifications in documents for a given subject. This new technique has been normally used in developing information recovery models.

In order to solve the above unsuitable statement, we present an effective pattern discovery technique, which evaluates discovered peculiarity of patterns. Then evaluate term weights according to the categorization of terms in the discovered patterns rather than the distribution, for correct the interpretation problem. It also considers the impact of patterns from the negative training to find ambiguous patterns and try to reduce their occurrences for the frequency problem. The process of updating doubtful patterns can be referred as pattern evolution. The proposed approach can improve the correctness of evaluating term weights because discovered patterns are more specific than whole documents.

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

2. Proposed Work

Presented system implement evaluation of term support based on their frequency in documents. Whereas the planned system is to fetch the information, that user wants in the form of terms or patterns. The data set or the data files are loaded into the data base. The data set goes for text pre-processing phase which has tokenization, parts of speech, and word stemming and stop word removal methods. Then terms present in the data set are examined in order to locate the positive documents and negative documents. In a second phase of Pattern taxonomy model, on positive data set the pattern deploying techniques were implemented. In third phase it discovers patterns. At the same time, negative sets suffer pattern evolving and shuffling techniques. Therefore the term supports are calculated. Terms will be fetched from term removal model.

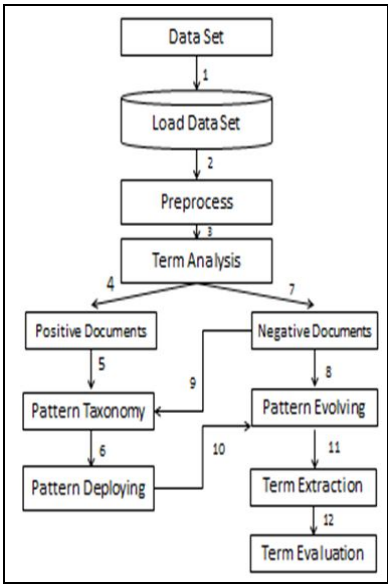

Figure 1: Proposed Architecture Model Flow

3. System Architecture

The proposed architecture is shown in Figure 1. Architecture shows the stepwise solution of our project. The basic step is to load SGML documents in our database. The next step is to convert that SGML document in XML format then remove stop word and text steaming. Removing that stop word and text steaming with the help of NLP (natural language process). Fig 1: System Architecture There is 5 sub modules of proposed system.

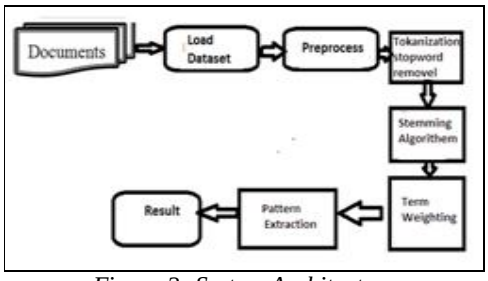

Figure 2: System Architecture

- Loading documents
- Text Preprocessing
- Term weighting
- Dimensionally Reduction
- Pattern Extraction

4. Document Selection

All of the tests run using the effective pattern mining rule for data processing. This tool will handle an oversized quantity of information. This part was important as a result of the dimensions of the log file for this project and it provided the specified algorithms for mining. The mining rules are effective pattern mining algorithm.

- Loading documents
- Text Pre-processing
- Term weight
- Dimensionally Reduction
- Pattern Extraction

4.1. Loading documents

In this module, to load the standard generalized markup language documents. The user retrieves one document. This document is given to next process that converts standard generalized markup language document to XML document. The conversion is necessary because standard generalized markup language document may be a code and it doesn't have any paragraph, the information keeps one after another. Thus it's necessary to convert it in text format.

Figure 3: loading document

The converted document has the sub roots like topic, date, place, people etc.

Figure 4: Converted SGML document to XML document

5. Pre-Processing

The pre-processing part of the process converts the prevailing matter information in an exceedingly data mining-ready structure. Wherever the necessary text-features that differentiates between text-categories which are known. Unstructured text documents processed using natural language processing techniques to remove keywords labeling, the things in this text document. An effective preprocessor describes the document efficiently in terms of each area (for storing the document) needs and support good retrieval performance. The main goal of pre-processing is to make the key options or key terms from text documents and to enhance the connection between word and document and therefore the connection between word and class.

Preprocessing technique contains.

- Text Clean Up
- Tokenization
- Stop word removal
- Word stemming

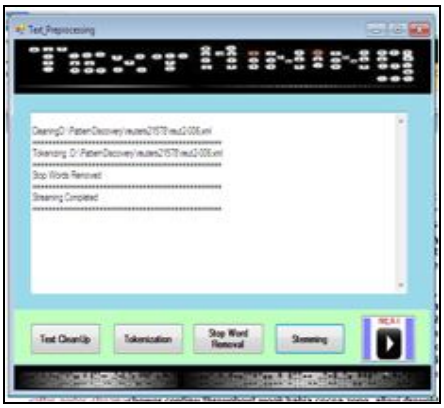

Figure 5: Pre-processing

5.1. Text Clean Up

After loading the document we see the XML document has several sub roots like date, people, org, exchange, companies, etc. In text cleanup process we will remove the roots that are unnecessary like people, org, exchange, people who is not having the meaning.

Figure 6: Text Clean Up document

5.2. Tokenization

Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. The list of tokens becomes input for further processing such as parsing or text mining. Tokenization is useful both in linguistics (where it is a form of text segmentation), and in computer science, where it forms part of lexical analysis.

Typically, tokenization occurs at the word level. However, it is sometimes difficult to define what is meant by a "word". Often a tokenize relies on simple heuristics, for example:

- All contiguous strings of alphabetic characters are part of one token; likewise with numbers
- Tokens are separated by whitespace characters, such as a space or line break, or by punctuation characters.
- Punctuation and whitespace may or may not be included in the resulting list of tokens.

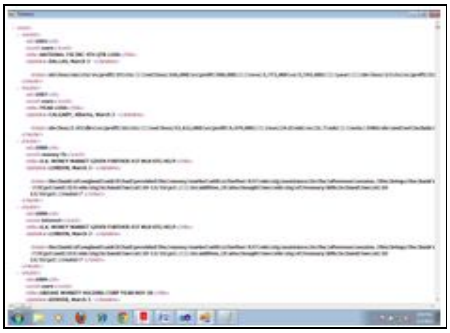

Figure 7: Tokenization of the document

5.3. Stop Word Removal

Many used words in English are worthless in Information Retrieval and text mining – these words are called stop words.

- Example the, of, and, to…
- Typically around 400 to 500 such words
- For an application, an additional area specific stop words list may be established

We remove stop words because Reduce indexing file size stops words accounts near 30% of total word counts. Stop words are not useful for searching or text mining so elimination of stop words improves efficiency, and stop words always get in a great number of hits.

In this step non informative words removed from the document, example the words like to, the, is, are, some that words removed from the text file.

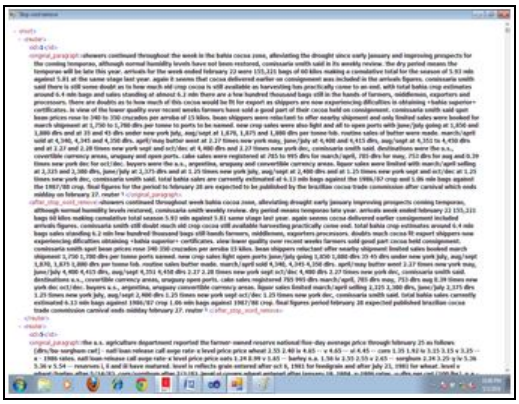

Figure 8: Stop Word Removal from the document

5.4. Parts of Speech

POS tagger marks the words during a text with labels similar to the part-of-speech of the word in this context. A part of speech tags the words consistent with the grammatical context of words by dividing it into nouns, verbs and a lot of. They're few representative tags for components of speech they're NN (single noun), NNS (plural noun), VB (verb), VBD (verb, past tense), VBN (verb, past participle), IN (preposition), JJ (adjective), CC (conjunction, e.g., "or", "and"), PRP (pronoun) MD (modal auxiliary, e.g., "will", "can").

5.5. Word Stemming

Stemming techniques wanted to resolve the root/stem of a word. Stemming converts words to their stems, which contains a good dependent linguistic data. Behind stemming, the assumption is that words with a similar meaning or word largely describe same or comparatively same ideas in text. Then the words will mix by using stems. The benefits of using stemming procedure is stemming improves effectiveness of knowledge Retrieval and text mining by matching similar words, largely improve recall. It reduces categorization size; hairdressing words with same roots might cut categorization size the maximum amount as 40-50%.

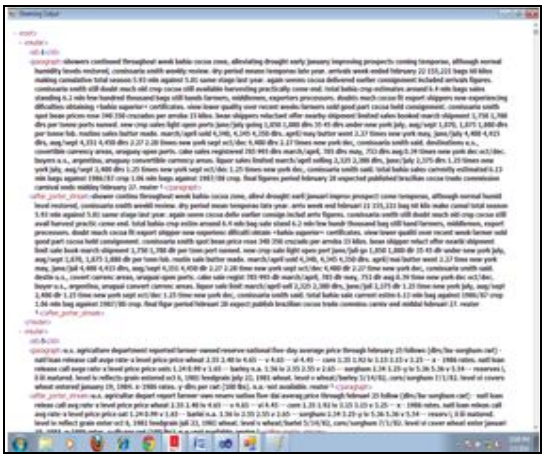

Figure 9: Result of Stemming Algorithm

6. Pattern Taxonamy Model

In this part, frequent patterns, sequential patterns, closed sequential pattern extracted. To enhance the potency of the pattern taxonomy mining, a formula, SP Mining, planned to search out all closed sequential patterns, which used the well-known Apriori property to split the leaking area. All the documents taken are split into paragraphs. Therefore a given document d yields a group of paragraphs ps (d). Let D be a coaching set of documents, which consists of a collection of positive documents, D+; and a collection of negative documents, D-. Let T= be a collection of terms (or keywords) which might extract from the set of positive documents, D+. Given a term set X in document d, ˹X˺ is employed to denote the covering set of X for d, which has all paragraphs dp€ PS(d) such that X is set of dp, i.e.,

˹X˺= {dp/dp € PS (d), X is set of dp}.

Frequent patterns discovery during this part using apriori rule so as to cut back the looking area for user. Frequent pattern discovery supported absolute support and relative support.

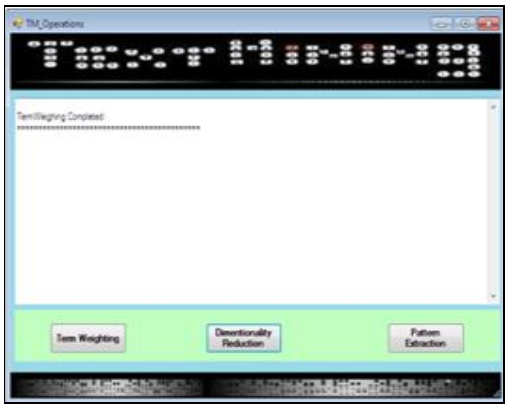

Figure 10: Taxonomy Model Operation

Absolute support is the quantity of occurrences of X in PS (d) denoted by supa(X) = |˹x˺|. Relative support is a fraction of the paragraphs that contain the pattern denoted by

supr(X) = |˹X˺|/|PS(d)|.

A term set X referred to as frequent pattern if its supr (or supa) min sup, a minimum support. Table one lists a collection of paragraphs for a given document d, wherever ps (d) =, and duplicate terms were removed.

Paragraph	Terms
dp1	t1 t2
dp2	t3 t4 t6
dp3	t3 t4 t5 t6
dp4	t3 t4 t5 t6
dp5	t1 t2 t6 t7

dp6	t1 t2 t6 t7

Table 1: Set of paragraphs

Let min sup = five hundredth, ten frequent patterns obtained in Table one using the on top of definitions. Table two illustrates the 10 frequent patterns and their covering sets.

Frequent patterns	Covering sets
$\{t_3, t_4, t_6\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3, t_4\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3, t_6\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_4, t_6\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3\}$	$dp_2, dp_3, dp_4\}$
$\{t_4\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_1, t_2\}$	$\{dp_1, dp_5, dp_6\}$
$\{t_1\}$	$\{dp_1, dp_5, dp_6\}$
$\{t_2\}$	$\{dp_1, dp_5, dp_6\}$
$\{t_6\}$	$\{dp_2, dp_3, dp_4, dp_5\}$

Table 2: Frequent patterns and covering sets

Not all frequent patterns in Table two are helpful. As an example, pattern perpetually happens with term t6 in paragraphs, i.e., the shorter pattern, , is usually a locality of the larger pattern, , altogether the paragraphs. Hence, the shorter one could be a noise pattern and expect to stay the larger pattern only.

Given a term set X, its covering set ˊXˋ could be a set of paragraphs. Similarly, given a group of paragraphs Y could be a set of PS(d), Its term set will defined, that satisfies

Term set(Y) = t Y => t dp} The closure of X is outlined as follows:

Cls(X) = term set (ˊXˋ).

A pattern X (also a term set) known as closed if and provided that X = Cls(X).

Let X is a closed pattern. We are able to prove that

Supa(X1) < supa(X),

For all patterns X could be a set of X1; otherwise, if supa(X1) = supa(X), we have

ˊX1ˋ = ˊXˋ

Where supa(X1) and supa(X) are absolutely the support ofpattern X1 and X severally.

Pattern taxonomy has evaluated to find closed patterns. Patterns are often structured into a taxonomy by using the (or subset) relation. For the instance of Table one, wherever a group of paragraphs of a document are illustrated, and also the discovered ten frequent patterns in Table two if assumptive min_sup =50%. There are, however, solely 3 closed patterns during this example. There, and Fig one illustrates associate example of the pattern taxonomy for the frequent patterns in Table two, wherever the nodes represent frequent patterns and their covering sets; non-closed patterns are often pruned; the perimeters are “is-a” relation. Once pruning, some direct “is-a” retaliations is also modified, as an example, pattern would become an on the spot sub pattern of once pruning non-closed patterns. From frequent patterns and closed patterns, closed successive patterns are discovered using SP Mining algorithmic program.

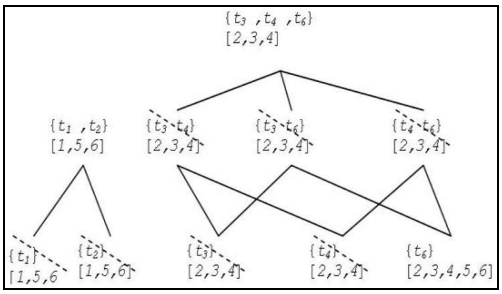

Figure 10: Pattern Taxonomy

6.1. Term-weighting

There are number of ways to define the term-weighting for the nonzero entries in such a vector. For instance, we will merely set TF (d, t) = one if the term t happens within the document d, or use the term frequency freq (d, t), or the relative term frequency, that is, the term frequency versus the whole variety of occurrences of all the terms within the document. There are alternative ways that to normalize the term frequency. For instance, the Cornell good system uses the subsequent formula to calculate the (normalized) term frequency:

$$TF(d,t) = \begin{cases} 0 & if\, freq(d,t) = 0 \\ 1 + \log(1 + \log(freq(d,t))) & otherwise. \end{cases}$$

Similarly the term frequency measure, in that place is another important quantity, called inverse document frequency (IDF) that represents the scaling element, or the importance, of a term t. If a term *t* occurs in many documents, its importance scaled down due to its reduced judgment power. For instance, the term database systems may likely be less significant if it occurs in many research papers in a database system conference.

According to Cornell good system, IDF (t) outlined by the subsequent formula:

$$IDF(t) = \log \frac{1 + |d|}{|d_t|},$$

Where d is the document collection, and dt is that the set of documents containing term.

Figure 11: Term Weighting

6.2. Dimensionally Reduction

In earlier term weighting we have the weights for every term or the token. Here in dimensionally reduction we will remove the terms or tokens who have weights less than user define and which is more than 0.

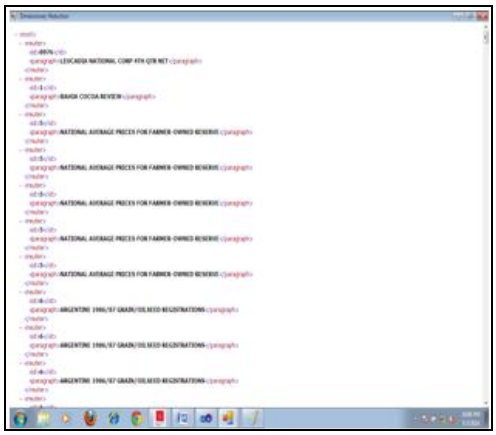

Figure 12: Dimensionally Reduction

6.3. Pattern Extraction

The searching process undergoes in frequent patterns, closed and closed sequential patterns in order to extract the information which a user wants in form of terms. Along with terms, nij is its support in di that is the total absolute paragraph (D+ or D-) extracted that has supports given by closed patterns that contain tij; or nij is total range of closed patterns that contain tij. The method of scheming d-patterns simply represented using the pattern taxonomy model formula. If the term what required has found

out during searching process in any of the extracted patterns, searching aborted. In this thesis for the support and confidence following formulas are used.

$$support(X \Rightarrow Y) = P(X \cup Y).$$
$$confidence(X \Rightarrow Y) = P(Y|X).$$

To this place support indicate that a group action has each X and Y, that is, union of item sets X and Y and confidence indicate conditional probability that's, the possibility that a transaction containing X also has Y.

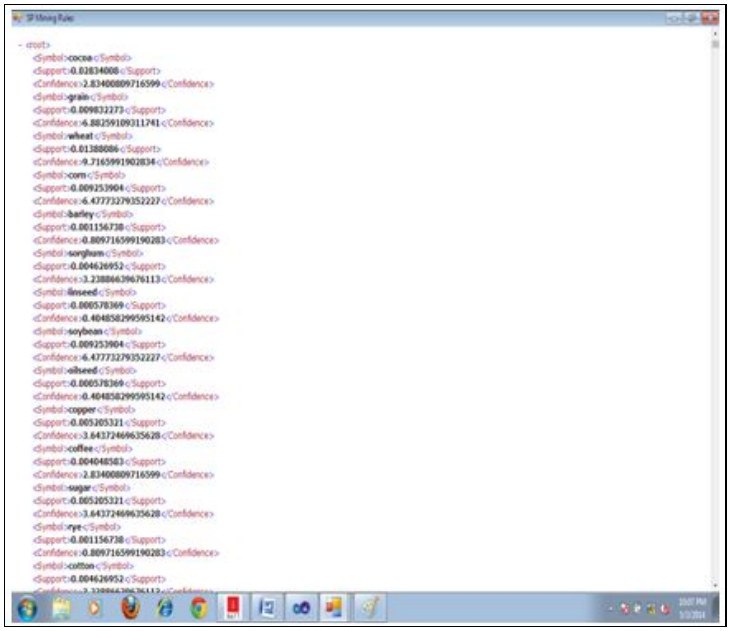

Figure 13: Pattern Extraction

7. Experimental Method

In the experimental procedure, few users allowed to choose the particular document what they want. In search option of this application developed, users enter the required keyword of their interest, such that selected document undergoes preprocessing, terms are analyzed, frequent patterns, closed and closed sequential patterns extracted. If the required key word obtained in any one of extracted patterns, searching process aborted. All these results will be analyzed along with the feedback taken from every user to know the accuracy of process. The time taken for obtaining the optimist result compared with time taken for term based approach. Hence it can be proved that planned approach is more correct than term based approach.

8 Conclusion

This new pattern discovery model for text mining mainly focuses on carry out temporal text pattern. Dynamic programming algorithm, knowledge and optimal lossless decomposition, introduced. This is used for analyzing the association between disintegration of time period linked with the document set and the important information computed for sequential analysis. It quickly finds the patterns for various ranges of the parameters. It focuses by means of data extraction to remove a structured database from a quantity of natural language text and then find out patterns in the resulting database using time-honored KDD tools. It also relates record linkage, a form of the data-cleaning that identifies equal but textually distinct items in the extracted data prior to mining. It is also related to the natural language learning.

9. References

1. T. Rose, M. Stevenson, and M. Whitehead, “The Reuters Corpus Volume1—From Yesterday’s News to Today’s Language Resources,” Proc. Third Int’l Conf. Language Resources and Evaluation, pp. 29-31, 2002.
2. M.F. Porter, “An Algorithm for Suffix Stripping,” Program, vol. 14, no. 3, pp. 130-137, 1980.
3. S.-T. Wu, Y. Li, and Y. Xu, “Deploying Approaches for Pattern Refinement in Text Mining” Proc. IEEE Sixth Int’l Conf. Data Mining (ICDM ’06), pp. 1157-1161, 2006.
4. 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.
5. Ning Zhong, Yuefeng Li, Sheng-Tang Wu ,Efficient Pattern Discovery for Text Mining, IEEE Transactions on Knowledge and Data Engineering, VOL. 24, NO. 1, JANUARY2012