ISSN:0975-9646
Vishakha D. Bhope et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (4) , 2015, 3625-3629

Information Retrieval using Pattern Deploying and Pattern Evolving Method for Text Mining

[1]Vishakha D. Bhope, [2]Sachin N. Deshmukh

[1,2]*Department of Computer Science & Information Technology,*
Dr. BAM University, Aurangabad

***Abstract*—Text mining is the process of extracting information from unstructured to structured text data. To mine user required information from text data in effective manner is a time consuming task. A variety of data mining techniques are available for information retrieval but it has drawbacks such as misinterpretation and low frequency of occurrence. Previously term based techniques were used which has drawback of having polysemy and synonymy words. This paper provides the information retrieval approach using pattern based method which uses pattern deploying and pattern evolving techniques.**

***Keywords*—Inner Pattern Evolution, Pattern Deploying Method, Sequential Pattern Mining**

I. INTRODUCTION

Text mining is the discovery of useful information from textual data. Most of the documents available on the web are in unstructured form, than in a structured form that can be automatically processed by a machine. It is a challenging task to retrieve relevant documents in response to a user's query from a huge document corpus, thereby satisfying the information need of the users [1].

Traditionally there are so many techniques available to solve the problems of text mining for retrieval of relevant information as per user's requirement. In text mining functions such as information extraction, categorization, text document analyzed on the basis of term [2], phrase [3], concept [4] and pattern.

Problem with term based approach is the semantic ambiguity which can be divided into synonymy and polysemy, where synonymy is multiple words having the same meaning and polysemy means a word has multiple meanings. Therefore for answering what users want, the semantic meaning of many discovered terms is uncertain [5].

To avoid the semantic ambiguity problem of term based methods new method is developed using multiple words (i.e. phrase) as a feature It is considered that the phrase based approaches could perform better than term based ones as more semantic information is carried by phrase than single term. But from several experiment results shown that the phrase based method is not superior to the term based method. Although phrases are less ambiguous and have more brief meaning than individual term, the likely reasons for the dispiriting performance include: 1) phrases have inferior statistical properties to terms, 2) low frequency of occurrence, and 3) large numbers of redundant and noisy phrases are present among them [6].

Instead of using the phrase-based and term-based methods efficient way for information retrieval is pattern based approach which contains frequent sequential patterns, because sequential patterns have good statistical properties like terms. To overcome the disadvantages of phrase based approaches, pattern taxonomy models have been proposed [7]. In the pattern taxonomy, semantic information will be used to improve the performance by using closed patterns in text mining. Sequential pattern mining concerned with finding relevant patterns in text dataset where values are delivered in sequence. There are two phases for using the pattern based models in text mining, first is how to discover useful patterns and other is how to utilize those patterns to improve system's performance [8].

The rest of this paper is structured as follows: Section II presents related work to the information retrieval approaches such as term-based, phrase-based, pattern based approach. Section III provides the proposed approach for Information Retrieval using Pattern-Based method. Section IV presents performance evaluation.

II. RELATED WORK

Many data mining techniques have been proposed for mining useful patterns in text documents. However, how to efficiently use and update discovered patterns is still an open research issue, specifically in the domain of text mining [5]. As most existing text mining methods adopted term-based approaches, they all suffer from the problems of polysemy and synonymy. From many years, people have often held the hypothesis that pattern and phrase-based approaches should perform better than the term-based ones, but many experiments do not hold up this hypothesis [6].

The choice of a representation depended on what one regards as the meaningful units of text and the meaningful natural language rules for the combination of these units [6]. For pattern based approaches there are two fundamental issues regarding the effectiveness of patterns: low frequency and misinterpretation. For a specified topic, a highly frequent pattern (which is normally a short pattern with large support) is usually a general pattern, or a specific pattern of low frequency. If the minimum support is decreased, a lot of noisy patterns would be discovered. Misinterpretation means the measures used in pattern

mining such as support turn out to be not suitable in using discovered patterns to answer what users want. Therefore the difficult problem is how to use discovered patterns to accurately evaluate the weights of useful features in text documents.

In text documents terms are important features, however many terms with larger weights are general terms because they can be frequently used in both relevant and irrelevant information. Therefore, it is not sufficient for evaluating the term weights based on their distributions in documents for a given topic. In order to solve above problem this paper presents an efficient pattern discovery technique. In this method term weight are calculated according to support of each term in the discovered patterns rather than the distribution of terms in whole document. This can solve the problem of misinterpretation. It also considers the influence of the patterns from the negative training set to find noisy terms and try to reduce their influence to solve the problem of low frequency. This process of updating noisy terms can be referred as Pattern Evolution. The pattern evolution technique was introduced in [9] in order to improve the performance of term-based ontology mining. The proposed approach can improve the accuracy of evaluating term weights because discovered patterns are more specific than whole documents. This section will also provide basic definition used in this system.

III. INFORMATION RETRIEVAL USING PATTERN BASED METHOD

The main objective of this work is to find the specific terms in the given input files using pattern based method. This method presents an effective solution for knowledge discovery technique, which can solve the problems like misinterpretation and low frequency of occurrence. The proposed approach can improve the accuracy of evaluating term weights because discovered patterns are more specific than whole documents. The following figure illustrate steps involved in finding the knowledge from text documents

A. Basic Definitions:

The basic definitions of sequence or pattern in this study are described as follow. Let T= {t1, t2, …, tk} be a set of all terms from each document, which can viewed as keyword in text dataset. A sequence S= <s1, s2, …., sn> (si ∈ T) is an ordered list of terms.

1) Absolute and Relative support

A termset X in document d, $\lceil X \rceil$ is used to denote the covering set of X for d, which includes all paragraphs dp ∈ PS(d) such that X ⊆ dp, i.e.,

$$\lceil X \rceil = \{dp | dp \in PS(d), X \subseteq dp\}$$

where dp is document paragraph and PS(d) is set of paragraphs of document d.

Its absolute support is the number of occurrences of X in PS(d) i.e.,

$$sup_a(X) = |\lceil X \rceil| \quad \text{(i)}$$

Its relative support is the fraction of the paragraphs that contain the pattern, that is,

$$sup_r(X) = \frac{|\lceil X \rceil|}{|PS(d)|} \quad \text{(ii)}$$

2) Frequent Sequential Pattern:

A termset X is called frequent pattern if its sup_r (or sup_a) ≥ min_sup, a minimum support. The purpose of using min_sup in this work is to reduce the number of patterns discovered in a large document. Otherwise the patterns with lower relative support will increase the burden of the training.

B. System Architecture:

Fig. 1 shows the system architecture of this knowledge discovery technique.

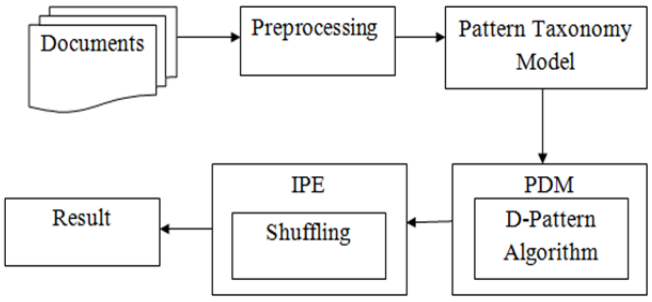

Fig. 1 System Architecture

Module Description:

1. Preprocessing
2. Pattern Taxonomy Model(PTM)
3. Pattern Deploying Method(PDM)
4. Inner Pattern Evolution(IPE)

1) Preprocessing

Preprocessing of text document is consists of removal of irrelevant data from documents. For textual data analysis less effective words are removed such as some verbs, pronouns, conjunctions, disjunctions, etc which are termed as stop words. Removal of these less informative words increases the efficiency and accuracy of results in processing the text.

2) Pattern Taxonomy Model(PTM)

PTM [5] is useful to know how to extract useful terms from text documents, and how to use these discovered terms to improve the effectiveness of a knowledge discovery system. Instead of previously used term-based

method for text representation model, the pattern based model is used which contains frequent sequential patterns (single term or multiple terms). The document set consists of positive (D+) and negative (D^-) documents. The documents are categorized by topic relevancy. Here all the text documents are splitted into set of paragraph; each paragraph consists of a set of words. Pattern Taxonomy is a pattern-based model for representing text documents. It is a Tree-like Structure which depicts out patterns being extracted from a text data. We discover a sequential pattern from collection of text documents and generate pattern taxonomy model to depict relationship between patterns extracted from the documents.

3) Pattern Deploying Method (PDM)

In this module to improve the performance of the pattern taxonomy method, the SPMining (Sequential Pattern Mining) is used to find out all frequent sequential patterns that uses the well-known Apriori property. The evaluation of term weights (supports) is different to the normal term-based approaches. In term-based approaches, the evaluation of term weights (supports) is based on the distribution of terms in documents. As suggested in [5], in deploying method, terms are weighted according to their appearances in discovered sequential patterns. It deploys patterns through the use of a pattern composition operator. The deploying method consists of the d-pattern discovery and term support evaluation [5]. Algorithm 1 is given below for D-Pattern Discovery:

Algorithm 1: D-Pattern Discovery

Input: positive documents D^+; minimum support, min_sup.
Output: d-pattern DP, support of terms.

1. DP=ø;
2. foreach document d∈D^+ do
3. let PS(d) be the set of paragraphs in d;
4. SP=SPMining(PS(d), min_sup);
5. dˆ= ø;
6. foreach pattern p∈ SP do
7. p={(t,1)|t∈p};
8. dˆ= dˆ⊕p;
9. end
10. DP=DP∪{ dˆ};
11. end
12. T={t|(t,f)∈p,p∈DP};
13. foreach term t∈T do
14. support(t)=0;
15. end
16. foreach d-pattern p∈DP do
17. foreach (t,w)∈β(p) do;
18. support(t)=support(t)+w;
19. end
20. end

In Algorithm 1 all discovered patterns in a positive document are composed into a d-pattern giving rise to a set of d-patterns DP in steps 6 to 9. Thereafter, from steps 12 to 19, term supports are calculated based on the normal forms for all terms in d-patterns.

For every positive document, the SPMining algorithm is first called giving rise to a set of frequent sequential patterns. Apriori is a basic principle used to improve the efficiency of sequential pattern mining.

Consider d = {(t1, w1), …….(ti, wi)}
Where w represents term support value for finding the sequence patterns in a given text documents. Here each individual pair representing the term and support value from the text documents. The termset taken as input includes the terms having term support larger than the min_sup.
Algorithm 2 is given below for Sequential pattern mining:

Algorithm 2: Sequential pattern Mining

Input: Sequence of terms
Output: Patterns (Combinations of terms)

1. Initialize output pattern to empty
2. foreach term in sequence from input start position to end of input term do
3. Append the term to the output pattern
4. Print generated patterns in output pattern
5. If current term is not the last in input sequence then
6. Generate remaining combinations starting at next position with iteration starting at next term beyond the term just selected.
7. Delete the last term of the output pattern
8. End
9. End

For each positive document d∈D+, possible set of patterns are discovered. After finding the available patterns in the document set, relative support is calculated for each pattern as given in equation (ii). By using the below formula we can calculate the D-patterns using the composition operation.

Let p_1 and p_2 are set of number of pairs of document

$p_1 \oplus p_2 = \{(t, x_1+x_2) \mid (t, x_1) \in p_1, (t, x_2) \in p_2\} \cup \{(t, x) \mid (t, x) \in p_1 \cup p_2, \text{not}((t, -) \in p_1 \cap p_2)\}$

where - is the wild card that matches any number.

For Example:

$\{(t1, 1), (t2, 2), (t3, 4)\} \oplus \{(t2, 3)\} = \{(t1,1), (t2, 5), (t3, 4)\}$

The process of calculating d-patterns is described by using the ⊕ operation in Algorithm 1 where a term's support is the total number of closed patterns that contain the term.

4) Inner Pattern Evolution (IPE)

Inner pattern evolution is implemented to reduce the side effects of noisy patterns which can solve low frequency problem. This method only changes the pattern's term supports within the pattern. To reduce the noise, it is necessary to track which d-patterns have been used to give rise to such an error. These patterns are called as offenders of negative document. There are two types of offenders: 1) A complete conflict offender which is a subset of negative document. These are removed firstly from d-patterns. 2) A partial conflict offender which contains part of terms of negative document. For these offenders their term supports are reshuffled to reduce the effects of noisy documents. The algorithm 2 for shuffling [5] is given below,

Algorithm 2: Shuffling

Input: a noised document nd, normal forms of d-patterns NDP, offenders Δ(nd), experimental coefficient μ.
Output: updated normal forms of d-patterns NDP

1. foreach d-pattern p in Δ(nd) do
2. if termset(p) ⊆ nd then NDP=NDP−{β(p)}; //remove complete conflict offenders
3. else //partial conflict offenders
4. $offering = (1-\frac{1}{\mu}) \times \sum_{t\in(termset(p)\cap nd)} support(t)$;
5. $base = \sum_{t\in(termset(p)-nd)} support(t)$;
6. Foreach tern t in termset(p) do
7. If t∈nd then $support(t) = (\frac{1}{\mu}) support(t)$; //shrink
8. Else //grow supports
9. $support(t) = support(t) \times (1 + offering \div base)$;
10. end
11. end

Shuffling algorithm is used to adjust the support distribution of terms within a d-pattern. The parameter offering is used for 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.

IV. PERFORMANCE EVALUATION

A popular text collection Reuters-21578 is used which has 21578 documents collected from the Reuters newswire. Among 90 categories, only the most populous 5 are used. 7911 documents are selected to use in our experiment, as shown in Table 1. Each category is employed as the positive examples class, and the rest as the negative examples class. This gives us 5 datasets.

TABLE 1: The most popular 5 categories on Reuters-21578 and their quantity

Acq	Crude	Earn	Money-fx	Wheat
2369	578	3964	717	283

1) Evaluation Measure

In our experiments, we use the popular F1 score on the positive examples class as the evaluation measure. F1 score takes into account of both recall and precision. Precision, recall and F1 defined as:

$$Precision = \frac{\#\ of\ relevant\ terms}{\#\ of\ retrieved\ terms}$$

$$Recall = \frac{\#\ of\ relevant\ terms}{\#\ of\ terms\ computed}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

For evaluating performance average across categories, macro-average is used. Macro-averaged performance scores are determined by first computing the performance measures per category and then averaging those to compute the global means. We use macro-averaging.

2) Experimental Results:
We have implemented this information retrieval technique using reuters-21578 collection taking five popular categories. In this method the relevant terms are found out from positive document set. Table 2 shows the results of this system.

TABLE 2: Performance Evaluation

Topic	Precision	Recall	F1
Acq	0.78	0.46	0.57
Crude	0.55	0.35	0.43
Earn	0.62	0.36	0.45
money-fx	0.6	0.4	0.48
Wheat	0.66	0.5	0.56

Macro average F1 score is 0.50. This measure is used to know the system's overall performance across sets of data.

V. CONCLUSION

Data mining techniques provides pattern mining methods but to use these patterns and update to solve misinterpretation and low frequency problem is achieved in this approach. Knowledge discovery with PDM and IPEvolving have been proposed to overcome the misinterpretation & low frequency problem. An effective knowledge discovery system is implemented using three main steps: (1) discovering useful patterns by sequential pattern mining algorithm (2) Using D- pattern discovery, term support evolution is done. (3) IPEvloving is used to reduce the influence of noisy terms. The experimental results show that the proposed model improves the performance of finding the accurate knowledge from the text data.

REFERENCES

[1] R. Sharma and S. Raman, "Phrase-Based Text Representation for Managing the Web Document," Proc. Int'l Conf. Information Technology: Computers and Comm. (ITCC), pp. 165-169, 2003.

[2] Yutaka Matsuo, Mitsuru Ishizuka "Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information," FLAIRS 2003.

[3] M.F. Caropreso, S. Matwin, and F. Sebastiani, "Statistical Phrases in Automated Text Categorization," Technical Report IEI-B4-07-2000, Instituto di Elaborazione dell'Informazione, 2000.

[4] S. Shehata, F. Karray, and M. Kamel, "A Concept-Based Model for Enhancing Text Categorization," Proc. 13th Int'l Conf. Knowledge Discovery and Data Mining (KDD '07), pp. 629-637, 2007.

[5] 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.

[6] F. Sebastiani. "Machine learning in automated text categorization," ACM Computing Surveys, 34 (1):1–47, 2002.

[7] 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.

[8] S. Shehata, F. Karray, and M. Kamel, "A Concept Model for Enhancing Text Categorization," Proc. 13th Int'l Conf. Knowledge Discovery and Data Mining (KDD '07), pp. 629-637, 2007.

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