Discovery of an Effective Pattern for Information Retrieval

[I]Tapaswini Nayak, [II]Prof (Dr). Srinivas Prasad, [III]Sumanjit Das
[I]PhD Scholar, Dept. of CSE, Centurion University of Technology and Management
[II]Professor, Dean (Academics), Dept. of CSE, GITA, BPUT.
[III]Asst. Professor, Dept. of CSE, Centurion University of Technology and Management

Abstract

Numerous 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 challenge, particularly 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. Substantial experiments on RCV1 data collection and TREC topics demonstrate that the proposed solution achieves encouraging performance.

Keywords

Information retrieval, Data mining, pattern mining, text mining.

I. Introduction

The field IR deals with the representation, storage of and access to information items. Modern IR is the most usual way of information access, mostly due to the increasing widespread of the World Wide Web (WWW). The concept of information in this context can be very misleading, as it is strongly bound to a user's request. IR systems are posed a user information need, normally in the form of a query, and they must return the whereabouts of pieces of information related to this enquiry, normally in the form of documents. IR systems present the information solely as a ranking of documents, whereas other systems, like Question-Answering (QA), might further elaborate this presentation with an ulterior process. The inherent subjective facet inside a user's interest implies that the problem of satisfying a user information need is always going to be open.

In this context, IR deals with the problem of finding and presenting documents of unstructured data that satisfy information need from within collections of documents. There are some points in the definition that need clarification. First of all, unstructured refers to data which have a semantically arbitrary structure, that can be conveyed unambiguously to a computer, as opposed to a relational database for instance. The term document refers to the granularity of the information presented to the user. It can represent abstracts, articles, Web pages, book chapters, emails, sentences, snippets, and so on. Moreover, the term document is not restricted to textual information, as users may be interested in retrieving and accessing multimedia data, like video, audio or images. Collection refers to a repository of documents from which information is retrieved. A user information need, also referred to as query, must be 'translated' in order for an IR system to process it and retrieve information relevant to its topic. This 'translation' is usually made by extracting a set of keywords that summaries the description of information need. Lastly, the presentation of the information must be in such a way that facilitates the user to find the documents that he is interested in. A good IR system must support document browsing and filtering tasks for facilitating a user's retrieval experience.

IR systems may be general or specialised with regard to the amount and type of data they have to scale up to, each facing different challenges. Examples of general IR systems are the widely available Web Search Engines (WSE) that process billions of heterogeneous Web documents. Examples of specialised IR systems are email retrieval systems or Desktop search systems integrated in modern Operating Systems (OS).

This issue of IR scalability is the main topic of interest of this paper IR system must provide responses to user queries efficiently, and they must do it fast. IR systems need to be fast because of the exponentially increasing amount of information that is available for retrieval today. In fact, today's technological advancements have allowed for vast amounts of information to be widely generated, disseminated, and stored. This has rendered the retrieval of relevant information a necessary and cumbersome task, which requires effective and efficient systems.

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 [4], rough set models [23], BM25 and support vector machine (SVM) [34] 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.

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

We ask that authors follow some simple guidelines. In essence, we ask you to make your paper look exactly like this document. The easiest way to do this is simply to download the template, and replace the content with your own material.

II. Information Retrieval Process

IR systems must cope with at least three different processes. The process of information retrieval is shown in fig-1

- Representing the content of documents.
- Representing a user’s information need.
- Comparing both representations.

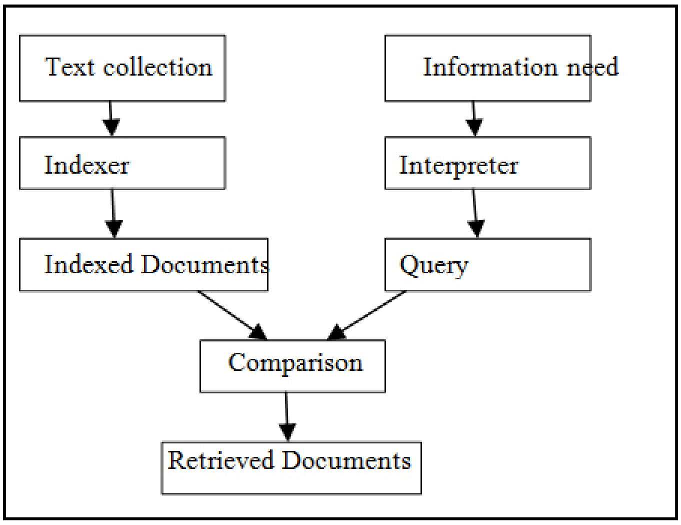


Fig 1: Process of information retrieval

The process of representing the content of documents is also called indexing.

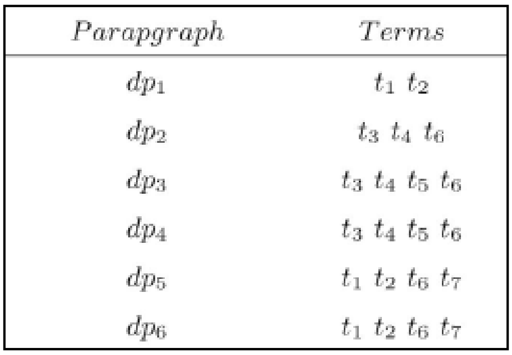
III. Related Work

In [3], data mining techniques have been used for text analysis by extracting co-occurring terms as descriptive phrases from document collections. However, the effectiveness of the text mining systems using phrases as text representation showed no significant improvement. The likely reason was that a phrase-based method had “lower consistency of assignment and lower document frequency for terms” as mentioned in [18].

Term-based ontology mining methods also provided some thoughts for text representations. For example, hierarchical clustering [28], [29] was used to determine synonymy and hyponymy relations between keywords. Also, the pattern evolution technique was introduced in [25] in order to improve the performance of term-based ontology mining.

For the challenging issue, closed sequential patterns have been used for text mining in [51], 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 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 [26] to significantly improve the performance of information filtering.

Table 1: A Set of Paragraphs



Parapgraph	*Terms*
dp_1	t_1 t_2
dp_2	t_3 t_4 t_6
dp_3	t_3 t_4 t_5 t_6
dp_4	t_3 t_4 t_5 t_6
dp_5	t_1 t_2 t_6 t_7
dp_6	t_1 t_2 t_6 t_7

IV. Pattern Taxonomy Model

In this paper, we assume that all documents are split into paragraphs. So a given document d yields a set of paragraphs PS (p). 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+.

Given a term set X in document d X’ is used to denote the covering set of X for d, which includes all paragraphs dp€ PS (d) such that X € dp i.e X’ {dp|dp € PS (d), X € dp}

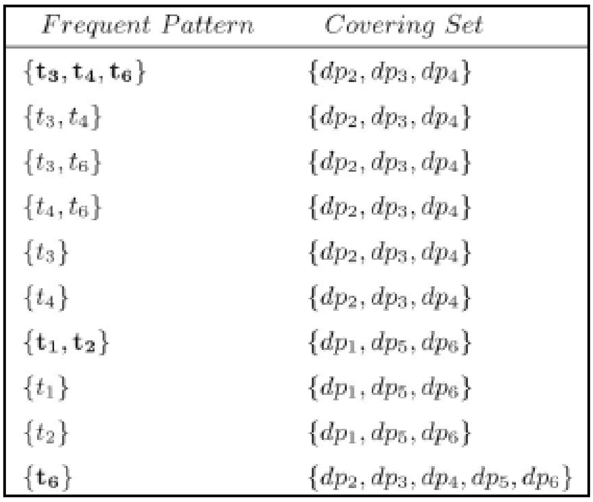
supr (X)=|X’|/ |PS(d)|.

A term X is called frequent pattern if it supr ≥ min_sup, a minimum support.

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

Not all frequent patterns in Table 2 are useful. For example, pattern ft3; t4g always occurs with term t6 in paragraphs, i.e., the shorter pattern, {t3; t4}, is always a part of the larger pattern, ft3; t4; t6g, in all of the paragraphs. Hence, we believe that the shorter one, ft3; t4g, is a noise pattern and expect to keep the larger pattern, {t3; t4; t6}, only.

Table 2: Frequent Patterns and covering sets.



Frequent Pattern	*Covering Set*
$\{\mathbf{t_3}, \mathbf{t_4}, \mathbf{t_6}\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3, t_4\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3, t_6\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_4, t_6\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_4\}$	$\{dp_2, dp_3, dp_4\}$
$\{\mathbf{t_1}, \mathbf{t_2}\}$	$\{dp_1, dp_5, dp_6\}$
$\{t_1\}$	$\{dp_1, dp_5, dp_6\}$
$\{t_2\}$	$\{dp_1, dp_5, dp_6\}$
$\{\mathbf{t_6}\}$	$\{dp_2, dp_3, dp_4, dp_5, dp_6\}$

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 three closed patterns in this example. They are <t3; t4; t6>, <t1; t2>, and <t6>.

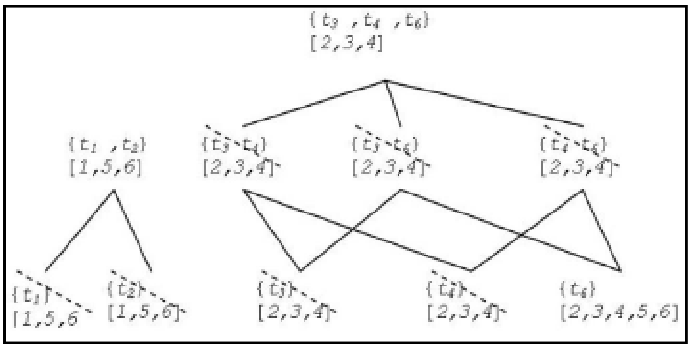


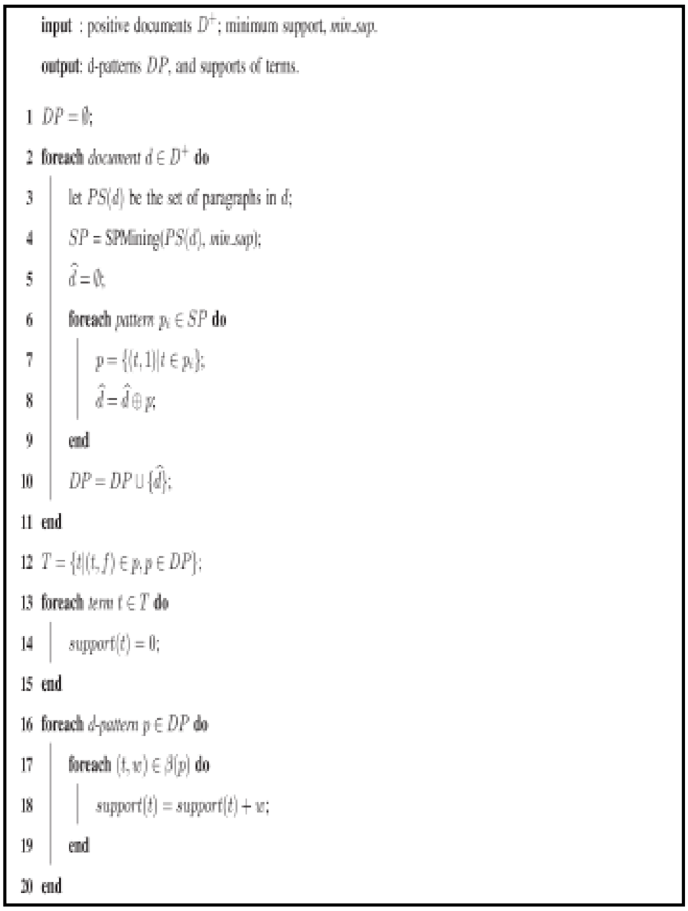
Fig. 2: illustrates an example of the pattern taxonomy for the frequent patterns in Table 2.

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 ft6g would become a direct subpattern of ft3; t4; t6g after pruning non closed 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. The semantic information will be used in the pattern taxonomy to improve the performance of using closed patterns in text mining, which will be further discussed in the next section.

V. Pattern Deploying Method

In order to use the semantic information in the pattern taxonomy to improve the performance of closed patterns in text mining, we need to interpret discovered patterns by summarizing them as d-patterns (see the definition below) in order to accurately evaluate term weights (supports). The rationale behind this motivation is that d-patterns include more semantic meaning than terms that are selected based on a term-based technique (e.g., tf*idf). As a result, a term with a higher tf*idf value could be meaningless if it has not cited by some d-patterns (some important parts in documents). The evaluation of term weights (supports) is different to the normal term-based approaches. In the term-based approaches, the evaluation of term weights is based on the distribution of terms in documents. In this research, terms are weighted according to their appearances in is covered closed patterns.

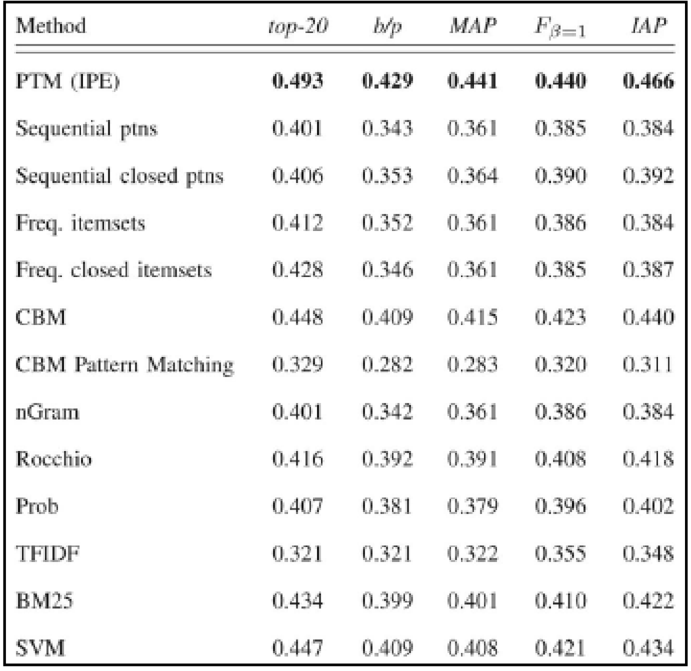


```
input : positive documents $D^+$; minimum support, $min\_sup$.
output: d-patterns $DP$, and supports of terms.

$DP = \emptyset$;
foreach document $d \in D^+$ do
    let $PS(d)$ be the set of paragraphs in $d$;
    $SP = \text{SPMining}(PS(d), min\_sup)$;
    $\hat{d} = \emptyset$;
    foreach pattern $p_i \in SP$ do
        $p = \{(t,1) | t \in p_i\}$;
        $\hat{d} = \hat{d} \oplus p$;
    end
    $DP = DP \cup \{\hat{d}\}$;
end
$T = \{t | (t,f) \in p, p \in DP\}$;
foreach term $t \in T$ do
    $support(t) = 0$;
end
foreach d-pattern $p \in DP$ do
    foreach $(t,w) \in \beta(p)$ do
        $support(t) = support(t) + w$;
    end
end
```

Fig. 3: Algorithm 1: PTM (D^+, min_sup).

Table 3: Comparison of All Methods on the First 50 Topics



Method	*top-20*	*b/p*	*MAP*	$F_{\beta=1}$	*IAP*
PTM (IPE)	**0.493**	**0.429**	**0.441**	**0.440**	**0.466**
Sequential ptns	0.401	0.343	0.361	0.385	0.384
Sequential closed ptns	0.406	0.353	0.364	0.390	0.392
Freq. itemsets	0.412	0.352	0.361	0.386	0.384
Freq. closed itemsets	0.428	0.346	0.361	0.385	0.387
CBM	0.448	0.409	0.415	0.423	0.440
CBM Pattern Matching	0.329	0.282	0.283	0.320	0.311
nGram	0.401	0.342	0.361	0.386	0.384
Rocchio	0.416	0.392	0.391	0.408	0.418
Prob	0.407	0.381	0.379	0.396	0.402
TFIDF	0.321	0.321	0.322	0.355	0.348
BM25	0.434	0.399	0.401	0.410	0.422
SVM	0.447	0.409	0.408	0.421	0.434

PTM (IPE) versus Other Models

The number of patterns used for training by each method is shown in Fig1. The total number of patterns is estimated by accumulating the number for each topic. As a result, the figure shows PTM (IPE) is the method that utilizes the least amount of patterns for concept learning compared to others. This is because the efficient scheme of pattern pruning is applied to the PTM (IPE) method.

Nevertheless, the classic methods such as Rocchio, Prob, and TFIDF adopt terms as patterns in the feature space; they use much more patterns than the proposed PTM (IPE) method and slightly less than the sequential closed pattern mining method. Particularly, nGram and the concept-based models are the methods with the lowest performance which requires more than 15,000 patterns for concept learning. In addition, the total number of patterns obtained based on the first 50 topics is almost the same as the number obtained based on the last 50 topics for all methods except PTM (IPE).

VI. Conclusions

There are many data mining techniques have been proposed in the last decade. These techniques include association rule mining, frequent item set 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 problemsfor text mining. The proposed technique uses two processes, pattern deploying and pattern evolving, to refine the discovered patterns in text documents. The experimental results show that the proposed model outperforms not only other pure data mining-based methods and the concept based model, but also term-based state-of-the-art models, such as BM25 and SVM-based models.

References

[1] *M. Zaki, "Spade: An Efficient Algorithm for Mining Frequent Sequences," Machine Learning, vol. 40, pp. 31-60, 2001.*

[2] *H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C.Watkins, "Text Classification Using String Kernels," J. Machine Learning Research, vol. 2, pp. 419-444, 2002.*

[3] *S.-T. Wu, Y. Li, Y. Xu, B. Pham, and P. Chen, "Automatic Pattern-Taxonomy Extraction for Web Mining," Proc. IEEE/ WIC/ACM Int'l Conf. Web Intelligence (WI '04), pp. 242-248, 2004.*

[4] *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.*

[5] *Y. Xu and Y. Li, "Generating Concise Association Rules," Proc. ACM 16th Conf. Information and Knowledge Management (CIKM '07), pp. 781-790, 2007.*

[6] *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.*

[7] *S. Shehata, F. Karray, and M. Kamel, "Enhancing Text Clustering Using Concept-Based Mining Model," Proc. IEEE Sixth Int'l Conf.Data Mining (ICDM '06), pp. 1043-1048, 2006.*

[8] *X. Yan, J. Han, and R. Afshar, "Clospan: Mining Closed Sequential Patterns in Large Datasets," Proc. SIAM Int'l Conf. Data Mining (SDM '03), pp. 166-177, 2003.*

[9] *Y. Li, X. Zhou, P. Bruza, Y. Xu, and R.Y. Lau, "A Two-Stage TextMining Model for Information Filtering," Proc. ACM 17th Conf.Information and Knowledge Management (CIKM '08), pp. 1023-1032, 2008.*