International Journal of Advanced Information in Engineering Technology (IJAIET) ISSN: 2454-6933
Vol.3, No.7, July 2016

Discovering Effective Patterns For An Efficient Document Clustering And Searching

Bhavya M
Dept. of Computer Science and Engineering
MEA Engineering College
Perinthalmanna, Kerala, India

Jemsheer Ahmed.
Dept. of Computer Science and Engineering
MEA Engineering College
Perinthalmanna, Kerala, India

Dr.Sobhana N V
Professor
Dept.of Computer Science and Engineering
RIT, Kottayam, Kerala, India

***Abstract*— Text mining has been an unavoidable data mining technique. There are different methods for text mining; one of the most successful will be mining using the effective patterns. 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 the process of document clustering and searching for finding relevant and interesting information.**

Index Terms— Text mining, Pattern discovery, Pattern Taxonomy

1. INTRODUCTION

As the World is being improvised in a digital way, knowledge discovery and Data mining have an important task. Useful information is always needed in all sort of information extraction. Text mining is therefore a step in knowledge discovery process in Databases and Datasets. Many data mining techniques have been proposed for mining useful patterns 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 existing, Information Retrieval (IR) provided many term-based methods to solve this challenge. The term-based methods suffer from the problems of polysemy and synonymy. Polysemy stands for a word having different meanings, and synonymy stands for different words having the same meaning. The Research Work use pattern (or phrase)-based approaches which perform better in comparison studies than other term-based methods. This approach improves the accuracy of evaluating support, term weights because discovered patterns are more specific than whole documents.

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 [13], 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 history of IR [5]. 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. In the presence of these set backs, sequential patterns used in data mining community have turned out to be a promising alternative to phrases [4], [12] because sequential patterns enjoy good statistical properties like terms. To overcome the disadvantages of phrase-based approaches, pattern mining-based approaches (or pattern taxonomy models (PTM) [12], [13]) have been proposed, which adopted the concept of closed sequential patterns, and pruned nonclosed patterns. These pattern mining-based approaches have shown certain extent improvements on the effectiveness.

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.

On the basis of obtained patterns clustering the documents and searching relevant data is more effective. In general, there are two common algorithms for document clustering. The first one is the hierarchical based algorithm, which includes single link, complete linkage, group average and Ward's method. By aggregating or dividing, documents can be clustered into hierarchical structure, which is suitable for browsing. However, such an algorithm usually suffers from efficiency problems. The other algorithm is developed using the K-means algorithm and its variants. These algorithms can further

be classified as hard or soft clustering algorithms. Hard clustering computes a hard assignment – each document is a member of exactly one cluster. The assignment of soft clustering algorithms is soft – a document's assignment is a distribution over all clusters. In a soft assignment, a document has fractional membership in several clusters.In this paper uses the K-means algorithm for clustering and then providing a searching option for retrieving information.

II. RELATED WORK

Pattern mining has been extensively studied in data mining communities for many years. A variety of efficient algorithms such as Apriori-like algorithms [11], PrefixSpan [9], FP-tree[1],[2], SPADE, SLPMiner[10], and GST [3] 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 [6], [7]. 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 [13], which proposed that the concept of closed patterns in text mining was useful and had the potential for improving the performance of text mining. After Pattern taxonomy model was also developed [12] 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.

Here we are proposing a pattern taxonomy model. Other different pattern mining methods are Sequential patterns, Sequential closed patterns, frequent itemsets, Frequent closed item sets. All these provide similar results but on depending on precision and recall our method stand way apart. The curves for PTM will remaining better and smoother when compared to the other pattern mining methods. When recall value raises the pattern mining methods starts coming down abruptly. This is shown in the figure.

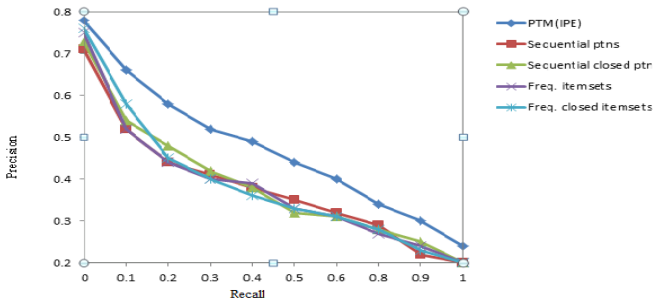

Fig.1: When recall value raises the pattern mining methods starts coming down abruptly

III. PROPOSED SYSTEM

A detailed description on the implementation aspects of the proposed work is discussed in this chapter.

A. System Modules

The entire work is divided into the following modules for better understanding of study and implementation.

1) Loading Document

As a first step this research work uses a collection of e-books, to evaluate the proposed technique.The user selects the document for processing.Selected document contents are then displayed. Input document contain both contents and tags. Here we are conducting the experiments on a set of e-books from different areas so as to provide a better clustering and searching result

2) Text Preprocessing

Pre-processing is a process of removing noise and incorrect data by data cleaning and data reduction techniques. Real-world database are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size often several gigabytes or more and their likely from multiple, heterogeneous sources. Low-quality data will lead to low-quality mining result, so it prefers a preprocessing concepts. The System perform preprocessing of text documents for the inputs are given to the PTM .The preprocessing has consists of two steps: Stop word removal and stemming process.

Stop word Removal:

Stop words are words which are filtered out prior to, or after, processing of natural language data. They typically comprise prepositions, articles, and so on. There is no specific list of stop words for all applications and these stop words are controlled by the human but not automated

Stemming Process:

Stemming is the process for reducing inflected (or sometimes derived) words to their stem base or root form. It generally a written word forms. In this preprocess the text documents have to be processed using the Porter stemmer. It removes the Suffix's of the words these words are useful in the text mining for clustering the text documents in the text mining process we collects the documents and each documents are composed into the set of terms or words .the words having stem have a same meaning .in stem process the suffixes of the words, singular and plural words are considered into a one single word for meaning full text clustering process.

3) Pattern Taxonomy Process

Pattern mining-based approaches or pattern taxonomy models (PTM) have been proposed, which adopted the concept of closed sequential patterns, and pruned nonclosed patterns. These pattern mining-based approaches have shown certain extent improvements on the effectiveness. However, the paradox is that people think pattern-based approaches could be a significant alternative, but consequently less significant improvements are made for the effectiveness compared with term-based methods. 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. On decreasing 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.
This research work assumes 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 and T = t1,t2,. . . ,tm be a set of terms.

Pattern Weight:

Formally, for all positive documents $d_i \in D+$, first deploy its closed patterns on a common set of terms T in order to obtain the following d-patterns (deployed patterns, nonsequential weighted patterns)

$d_i = \{(t_{i1},n_{i1}) , (t_{i2},n_{i2}),\ldots(t_{im}, n_{im})\}$

Where t_{ij} in pair ($t_{ij,}$ nij) denotes a single term and n_{ij} is its support in d_i which is the total absolute supports given by closed patterns that contain t_{ij}; or n_{ij} is the total number of closed patterns that contain t_{ij}.

Term Support:

Given a termset X in document d, ⌜X⌝ is used to denote the covering set of X for d, which includes all paragraphs dpεPS(d) such that

$$⌜X⌝=\{dp|dp \in Ps(d), X \subseteq dp\}$$

Its absolute support is the number of occurrences of X in PS (d), that is $sup_a(X) = |⌜X⌝|$. Its relative support is the fraction of the paragraphs that contain the pattern. A termset X is called frequent pattern if its sup_r (or sup_a) >= min_sup, a minimum support.

Closed Sequential Patterns:

A sequential pattern s = <t1; . . . ; tr> (ti∈ T) is an ordered list of terms. A sequence s1 = <x1; . . . ; xi> is a subsequence of another sequence s2 = <y1; . . . ; yj>, denoted by s1 ∈ s2, iff ɔ j_1; . . . ; j_y such that $1 \leq j1 < j2 \ldots <jy \leq j$ and x1 = yj_1; x2 = yj_2 ; . . . ; xi = yj_y. Given s1 ∈ s2, usually say s1 is a subpattern of s2, and s2 is a super pattern of s1. In the following, simply say patterns for sequential patterns.
Given a pattern (an ordered termset) X in document d, ⌜X⌝ is still used to denote the covering set of X, which includes all paragraphs ps∈PS(d) such that X € ps, i.e., ,⌜X⌝ = {ps|ps∈ PS(d), X € ps}. Its absolute support is the number of occurrences of X in PS(d), that is $sup_a(X) = |⌜X⌝|$. Its relative support is the fraction of the paragraphs that contain the pattern. A sequential pattern X is called frequent pattern if its relative support (or absolute support) ≥ min_sup, a minimum support. The property of closed patterns can be used to define closed sequential patterns. A frequent sequential pattern X is called closed if not ɔ any super pattern X1 of X such that $sup_a(X1) = sup_a(X)$.

4) Pattern Deploying

In order to use the semantic information in the pattern taxonomy to improve the performance of closed patterns in text mining. The rational 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 are based on the distribution of terms in documents. In this research work, terms are weighted according to their appearances in discovered closed patterns. It is complicated to derive a method to apply discovered patterns in text documents for information filtering systems.The discovered patterns are summarized. The d-pattern algorithm is used to discover all patterns in positive documents are composed. The term supports are calculated by all terms in d-pattern. Term support means weight of the term is evaluated. 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. 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 D pattern, all discovered patterns 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 are calculated based on the normal forms for all terms in dpatterns.

Algorithm 1: d-pattern Mining Algorithm

Input:
Positive documents D^+; minimum support, min_sup.
Output:
d-patterns DP and support of terms.

Steps:
1. DP= Ø
2. for each document d ϵ D^+ do
3. let PS(d) be the set of paragraphs in d;
4. SP = SPMining(PS (d), min_sup);
5. d=Ø
6. for each pattern p_i ϵ SP do
7. p = {(t, 1)| t ϵ p_i };
8. d =d+p
9. end
10. DP=DP U{d};
11. end
12. T = {(t| (t, f)ε p, p ϵ DP};
13. for each term tϵ T do
14. support (t) =0;
15. end
16. for each d-pattern p ϵ DP do
17. for each(t,w) ϵ β(p)do
18. support(t)= support (t) + w;
19. end
20. end

5) Pattern Evolving

In this section, discusses 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. Some documents in D is a negative document that the system falsely identified as a positive, In order to reduce the noise, The research work track which d-patterns have been used to give rise to such an error. These patterns are called offenders.
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 are reshuffled in order to reduce the effects of noise documents. The main process of inner pattern evolution is implemented by the algorithm IPEvolving.

Inner Pattern Evolution

This section, discuss about the shuffling of supports of terms d-patterns based on negative documents in the training set. This reduces the side effects of noisy pattern because of the low-frequency problem. It changes only a pattern's term supports within the pattern, this technique is called inner pattern evolution. Documents into relevant or irrelevant categories based on a Threshold. The main process of inner pattern evolution is implemented by the algorithm IP Evolving .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-patterns.Step 2 estimates the threshold for finding the noise negative documents. Thereafter Steps 3 to 10 go over term supports by using all noise negative documents. Step 4 is for finding the noise documents. Step 5 gets normal forms of dpatterns NDP. Step 6 calls algorithm shuffling to update NDP according to noise documents. Thereafter Steps 7 to 9 binds the updated normal forms together

Algorithm 2: IP Evolving Algorithm

Input:
A training set D=D+U D-,a set of D patterns DP and an experimental coefficient μ
Output:
A set of term support pairs np

Steps:
1.Np ←φ
2.Threshold= Threshold(DP)
3.Foreach noise negative document nd Є D- do
4.If weight(nd)≥threshold then,Δ(nd)={p element DP |termset (p) ∩ nd ≠φ};
5.NDP={β(p)|p Є DP};
6.Shuffling (nd, ,Δ(nd), NDP,μ NDP);// call Alg 3:
7.for each p Є NDP do;
8.np ← np Θ p;
9.end
10.end

Shuffling

In algorithm 3 the parameter offering is used in step 4 for the purpose of storing the reduced supports temporarily of some terms in a partial conflict offender. Here the offering is part of the sum of supports of terms in a d-pattern where these terms also appear in a noise document. Thereafter the algorithm calculates the base in step 5 which is non-zero .The updation of the support distributions of terms is done in step 6.

Algorithm 3: Shuffling Algorithm

Input:
A noise document nd; its offenders*Δ (*nd); normal patterns of D patterns NDP and an experimental coefficient μ.

Output:
Updated normal forms of d- patterns NDP.

Steps:
1. for each d-pattern p in Δ(nd) do
2. if term set (p) ⊆ nd; then NDP= NDP-{β (p)};
3. else partial conflict offenders
4. Offering = (1-1/μ) ×∑ support (t)
5. base = ∑ support(t)
6. for each term t in term set (p) term t do
7. if t ϵ nd then support(t) = 1/μ×support (t);// shrink
8. else // grow supports
9. support(t) = support(t) × (1 + offering ÷ base);
10.end
11. end

6) *Clustering Documents*

After extracting deployed patterns, perform the k-means clustering process to cluster the documents based on these patterns. So in this system these deployed patterns are plays an important role in the clustering process. The K-Means clustering algorithm is a partition-based cluster analysis method. According to the algorithm we first select k objects as initial cluster centers, then calculate the distance between each cluster centre and each object and assign it to the nearest cluster, update the averages of all clusters ,repeat this process until the criterion function converged.

Algorithm 4: K-means Clustering Algorithm

Input:
N objects to be cluster {x_1, x_2.....x_n}, the number of clusters k.

Output:
k clusters and the sum of dissimilarity between each object and its nearest cluster center is the smallest.

Steps:
1. Arbitrarily select k objects as initial cluster centers (m_1, m_2... m_k);

2. Calculate the distance between each object x_i and each cluster center, then assign each object to the nearest cluster, formula for calculating distance as:

$$d(x_i, m_i) = \sqrt{\sum_{j=1}^{d} (x_{i1} - m_{j1})}$$

i = 1,2.....N
j = 1,2.....k
d (x_i, m_j) is the distance between data i and cluster j.
3.Calculate the mean of objects in each cluster as the new cluster centers,

$$m_i = \frac{1}{N_i} \sum_{j=1}^{N_i} x_{ij}$$

i=l, 2 . . . k; N_i is the number of samples of current cluster i.

7) Searching Query

After the clustering process the user queries are searching from the patterns obtained .Cluster based query searching and whole searching is performed here and a comparison of both also.

IV. RESULTS AND OBSERVATIONS

An effective pattern discovery technique is discovered. Evaluates specificities of patterns and then evaluates term weights according to the distribution of terms in the discovered patterns Solves Misinterpretation Problem. Considers the influence of patterns from the negative training examples to find ambiguous (noisy) patterns and tries to reduce their influence for the low-frequency problem. The process of updating ambiguous pattern scan be referred as pattern evolution. The proposed approach can improve the accuracy of evaluating term weights because discovered patterns are more specific than whole documents. In General there are two phases Training and Testing In training phase the d-patterns in positive documents (Dp) based on a min sup are found, and evaluates term supports by deploying dpatterns to terms.In Testing Phase to revise term supports using noise negative documents in D based on an experimental coefficient the incoming documents then can be sorted based on these weights. 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 and clustering the documents on the basis of these patterns helps users for finding relevant and interesting information. The proposed work has the merits including 1) used to improve the accuracy of evaluating term weights. Because, the discovered patterns are more specific than whole documents.2) To avoiding the issues of phrase-based approach using the pattern-based approach.3) Pattern mining techniques can be used to find various text patterns.4) Execution time is minimized.5) Complexity of time is reduced 6) Clustering results has more accuracy than other document clustering system.

V. CONCLUSION AND FUTURE WORK

The research area is an emerging technology. Thus doing research in this area explores new area of study with more scope. 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).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. Clustering is the division of data into groups of similar objects. In clustering, some details are disregarded in exchange for data simplification. Clustering can be viewed as a data modeling technique that provides for concise summaries of the data. Clustering is therefore related to many disciplines and plays an important role in a broad range of applications. The applications of clustering usually deal with large datasets and data with many attributes. In the existing system-means clustering algorithm is performed or the whole data. But in this system there are many issues. One of the issues is time complexity of the clustering. To improve the performance and reduce the time complexity of the clustering process we introduce in the proposed system effective pattern discovery. technique. 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. Our proposed document clustering approach is well effective among the other document clustering techniques. Uncertainties can arise at any stage of a pattern classification system, resulting from incomplete or imprecise input information, ambiguity or vagueness in input data, ill defined and/or overlapping boundaries among classes or regions, and indefiniteness indefining/extracting features and relations among them. It is therefore necessary for a classification system to have sufficient provision for representing uncertainties involved at every stage so that the final output (results) of the system is associated with the least possible uncertainty. Another approach is to develop a search engine based on the proposed system that is an efficient pattern based search engine.

REFERENCES

[1] J. Han and K.C.-C. Chang, “Data Mining for Web Intelligence,”Computer, vol. 35, no. 11, pp. 64-70, Nov. 2002.
[2] J. Han, J. Pei, and Y. Yin, “Mining Frequent Patterns without Candidate Generation,” Proc. ACM

SIGMOD Int'l Conf. Management of Data (SIGMOD '00), pp. 1-12, 2000.

[3] Y. Huang and S. Lin, "Mining Sequential Patterns Using Graph Search Techniques," Proc. 27th Ann. Int'l Computer Software and Applications Conf., pp. 4-9, 2003.

[4] N. Jindal and B. Liu, "Identifying Comparative Sentences in Text Documents," Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '06), pp. 244-251, 2006

[5] D.D. Lewis, "Feature Selection and Feature Extraction for Text Categorization," Proc. Workshop Speech and Natural Language,pp. 212-217, 1992

[6] Y. Li, W. Yang, and Y. Xu, "Multi-Tier Granule Mining for Representations of Multidimensional Association Rules," Proc. IEEE Sixth Int'l Conf. Data Mining (ICDM '06), pp. 953-958, 2006.

[7] 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

[8] J.S. Park, M.S. Chen, and P.S. Yu, "An Effective Hash-Based Algorithm for Mining Association Rules," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '95), pp. 175-186, 1995

[9] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "Prefixspan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth," Proc. 17th Int'l Conf. Data Eng. (ICDE '01), pp. 215-224, 2001.

[10] M. Seno and G. Karypis, "Slpminer: An Algorithm for Finding Frequent Sequential Patterns Using Length-Decreasing Support Constraint," Proc. IEEE Second Int'l Conf. Data Mining (ICDM '02), pp. 418-425, 2002

[11] R. Srikant and R. Agrawal, "Mining Generalized Association Rules," Proc. 21th Int'l Conf. Very Large Data Bases (VLDB '95),pp. 407-419, 1995.

[12] 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.

[13] S.-T. Wu, Y. Li, Y. Xu, B. Pham, and P. Chen, "Automatic Pattern-Taxonomy Extraction for Web Mining," Conf. Web Intelligence (WI '04), pp. 242-248, 2004