RESEARCH ARTICLE OPEN ACCESS

Text Mining For Information Filtering Using Patterns

Srujini.T, Deepthi.B
(Assistant professor, Dept. Of CSE, Jawaharlal Nehru Institute of technology, Hyderabad, India
(Assistant professor, Dept. Of CSE, Jawaharlal Nehru institute of technology, Hyderabad, India

Abstract— Many data mining techniques have been proposed with regard to mining valuable patterns with text docs. However, how you can effectively utilize and update discovered patterns is still an start research problem, especially from the domain regarding text mining. Since nearly all existing text message mining methods adopted term-based methods, they all experience the problems of polysemy and also synonymy. Over the years, people include often used the hypothesis that style (or phrase)-based methods should perform better than the term-based people, but a lot of experiments don't support this hypothesis. This project presents a progressive and effective pattern discovery technique which includes the techniques of style deploying and also pattern changing, to improve the potency of using and also updating observed patterns with regard to finding applicable and fascinating information.

Keywords— Classification, Granule mining, Pattern Mining, Text Mining,

I. INTRODUCTION

Many types of text representations have been proposed in the past. A well known one is the bag of words that uses keywords (terms) as elements in the vector of the feature space. The tf/idf weighting scheme is used for text representation in Rocchio classifiers. In addition to TFIDF, the global IDF and entropy weighting scheme is proposed and improves performance by an average of 30 percent. Various weighting schemes for the bag of words representation approach were proposed. The problem of the bag of words approach is how to select a limited number of features among an enormous set of words or terms in order to increase the system's efficiency and avoid over fitting. In order to reduce the number of features, many dimensionality reduction approaches have been conducted by the use of feature selection techniques, such as Information Gain, Mutual Information, Chi-Square, Odds ratio, and so on. 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. With respect to the representation of the content of documents, some research works have used phrases rather than individual words. The combination of unigram and bigrams was chosen for document indexing in text categorization (TC) and evaluated on a variety of feature evaluation functions (FEF). A phrase-based text representation for Web document management was also proposed.

In order to solve the above paradox, this project presents an effective pattern discovery technique, which first calculates discovered specificities of patterns and then evaluates term weights according to the distribution of terms in the discovered patterns rather than the distribution in documents for solving the misinterpretation problem. It also considers the influence of patterns from the negative training examples to find ambiguous (noisy) patterns and try to reduce their influence for the low-frequency problem. The process of updating ambiguous patterns can 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. 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

II. PREVIOUS WORK

Corpus-based supervised learning is now a standard approach to achieve high-performance in natural language processing. However, the weakness of supervised learning approach is to need an annotated corpus, the size of which is reasonably large. Even if we have a good supervised-learning method, we cannot get high-performance without an annotated corpus. The problem is that corpus annotation is labor intensive and very expensive. In order to overcome this, several methods are proposed, including minimally-supervised learning methods and active learning methods. The spirit behind these methods is to utilize precious labeled examples maximally.

Another method following the same spirit is one using *virtual examples* artificially created examples generated from labeled examples. This method has been rarely discussed in natural language processing. In terms of active learning, Lewis and Gale mentioned the use of virtual examples in text classification. They did not, however, take forward this approach because it did not seem to be possible that a classifier created virtual examples of documents in natural language and then requested a human teacher to label them.

In the field of pattern recognition, some kind of virtual examples has been studied. The first report of methods using virtual examples with Support Vector Machines (SVMs), who demonstrated significant improvement of the accuracy in hand-written digit recognition. They created virtual examples from labeled examples based on prior knowledge of the task: slightly translated images have the same label (class) of the original image. Also discussed the use of prior knowledge by creating virtual examples and thereby expanding the effective training set size. [1]

Comparisons are one of the most convincing ways of evaluation. Extracting comparative sentences from text is useful for many applications. For example, in the business environment, whenever a new product comes into market, the product manufacturer wants to know consumer opinions on the product, and how the product compares with those of its competitors. Much of such information is now readily available on the Web in the form of customer reviews, forum discussions, blogs, etc. Extracting such information can significantly help businesses in their marketing and product benchmarking efforts. In this paper, we focus on comparisons. Clearly, product comparisons are not only useful for product manufacturers, but also to potential customers as they enable customers to make better purchasing decisions.

In the past few years, a significant amount of research was done on sentiment and opinion extraction and classification. In the existing literature and compare it with our work, where related research from linguistics is also included. Comparisons are related but also quite different from sentiments and opinions, which are subjective. Comparisons on the other hand can be subjective or objective. For example, an opinion sentence on a car may be "*Car X is very ugly*". A subjective comparative sentence may be "*Car X is much better than Car Y*" An objective comparative sentence may be "*Car X is 2 feet longer than Car Y*"

We can see that in general comparative sentences use quite different language constructs from typical opinion sentences although the first sentence above is also an opinion. In this paper, we aim to study the problem of identifying comparative sentences in text documents, e.g., news articles, consumer reviews of products, forum discussions. This problem is challenging because although we can see that the above example sentences all contain some indicators (comparative adverbs and comparative adjectives), i.e., "better", "long*er*", many sentences that contain such words are not comparatives, e.g., "*I cannot agree with you more*". Similarly, many sentences that do not contain such indicators are comparative sentences, e.g., "*Cellphone X has Bluetooth, but cellphone Y does not have.*" [2]

Related work to ours comes from both computer science and linguistics. Researchers in linguistics focus primarily on defining the syntax and semantics of comparative constructs. They do not deal with the identification of comparative sentences from a text document computationally. Studies the semantics and syntax of comparative sentences, but uses only limited vocabulary. It is not able to do our task of identifying comparative sentences. Discusses gradability of comparatives and measure of gradability. The semantic analysis is based on logic, which is not directly applicable to identifying comparative sentences. The types of comparatives such as adjectival, adverbial, nominal, superlatives, etc are described. The focus of these researches is on a limited set of comparative constructs which have gradable keywords like more, less, etc. In summary, although linguists have studied comparatives, their semantic analysis of comparatives based on logic and grammars is more for human consumption than for automatic identification of comparative sentences by computers.

In text and data mining, we have not found any direct work on comparative sentences. The most closely related work is sentiment classification and opinion extraction, which as we pointed out in the introduction section are related but quite different from our work. Sentiment classification classifies opinion texts or sentences as positive or negative. Work of Hearst on classification of entire documents uses models inspired by cognitive linguistics. Das and Chen use a manually crafted lexicon in conjunction with several scoring methods to classify stock postings. Tong generates sentiment positive and negative timelines by tracking online discussions about movies over time. Applies a unsupervised learning technique based on mutual information between document phrases and the words "excellent" and "poor" to find indicative words of opinions for classification examines several supervised machine learning methods for sentiment classification of movie reviews. Also experiments a number of learning methods for review classification. They show that the classifiers perform well on whole reviews, but poorly on sentences because a sentence contains much less information. Investigates sentence subjectivity classification. A method is proposed to find adjectives that are indicative of positive or negative opinions. Proposes a similar method for nouns. Other related works on sentiment classification and opinions discovery include. In several unsupervised and supervised techniques are proposed to analyze opinions in customer reviews. Specifically, they identify product features that have been commented on by customers and determining whether the opinions are positive or negative.

However, none of these studies is on comparison, which is the focus of this work. [3]

Multidimensional association mining discusses two or more data dimensions or predicates. Usually multidimensional association mining is designed for searching frequent predicate sets and that can be classified into inter-dimension and hybrid-dimension association rule mining. We can obtain a huge amount of association rules using the existing data mining techniques. However, not all strong association rules are interesting to users. Several approaches have been conducted in order to guarantee the quality of discovered knowledge: the concept of closed patterns, non-redundant rules, and constraint-based association rules.

These approaches have significant performance for decreasing the number of association rules for transaction databases. However, they are not very efficient for representation of associations in very large multidimensional databases because we have to transfer multidimensional rule mining into single dimensional mining when we use these approaches. Different to these approaches, in this paper we present the concept of granule mining (GM) in multidimensional databases for directly representations of associations between attributes, where a granule is a group of objects (transactions) that have the same attributes' values.

Basically attributes are divided by users into two groups: condition attributes and decision attributes, and decision tables can be used to represent the association between condition granules and decision granules. In cases of large number of attributes, however, decision tables become inefficient. Decision tables also cannot describe association rules with shorter premises. To solve these drawbacks, in this paper we present multi-tier structures and association mappings to manage associations between attributes. It provides an alternative way to represent multidimensional association rules efficiently. [4]

The application of machine learning techniques to classification of documents is a rich and challenging research area with many related tasks, such as routing, filtering or cross-lingual information retrieval. Since Joachims and other researchers like Yang and Liu have shown that Support Vector Machines (SVM) perform favourably compared to competing techniques for document categorisation, kernel machines have been a popular choice for document processing. In most reported works, however, documents were represented using the standard vector space, aka *bag-of-word* model that is, more or less, word frequencies with various added normalizations in conjunction with general purpose kernels.

Recently, Watkins and Lodhi et al. proposed the use of *string kernels*, one of the first significant departures from the vector space model. In string kernels, the features are not word frequencies or an implicit expansion thereof, but the extent to which all possible ordered subsequences of characters are represented in the document. In addition, Lodhi et al. proposed a recursive dynamic programming formulation allowing the practical computation of the similarity between two sequences of arbitrary symbols as the dot-product in the implicit feature space of all ordered, non-consecutive subsequences of symbols. Although it allows to perform the kernel calculation without performing an explicit feature space expansion, this formulation is extremely computationally demanding and is not applicable with current processing power to large document collections without approximation.

In this document, we propose to extend the idea of sequence kernels to process documents as sequences of words. This greatly expands the number of symbols to consider, as symbols are words rather than characters, but it reduces the average number of symbols per document. As the dynamic programming formulation used for computing sequence matching depends only on sequence length, this yields a significant improvement in computing efficiency. Training a SVM on a dataset of around 10000 documents like the Reuters-21578 corpus becomes feasible without approximation. In addition, matching sequences of words allows working with symbols that are expected to be more linguistically meaningful. This leads to extensions of the word sequence kernels that implement a kind of inverse document frequency (IDF) weighting by allowing symbol varying decay factors. Words may also be equivalent in some context, and we show how to implement soft word matching in conjunction with the word-sequence kernel. [5]

III. PROPOSED SYSTEM

3.1. Frequent and Closed Patterns

In this module 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 belonging to Paragraph Set. Its absolute support is the number of occurrences of X in PS. Its relative support is the fraction of the paragraphs that contain the pattern, that is, supr. A term set X is called frequent pattern if its supr or supa is greater than or equal to a minimum support. The duplicate terms were removed. All the Frequent patterns may not be useful, hence, we believe that the shorter one is a noise pattern and expect to keep the larger pattern only. Given a term set X, its covering set X' is a subset of paragraphs. Similarly, given a set of paragraphs PS we can define its term set. The closure of X is defined. A pattern X also a term set is called closed if and only if X is closed. Patterns can be structured into a taxonomy by using the is-a (or subset) relation, 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. Smaller patterns in the taxonomy are usually more general because they could be used frequently in both positive and negative documents; and larger patterns. The semantic information will be used in the pattern taxonomy to improve the performance of using closed patterns in text mining.

3.2. Closed Sequential Patterns

In this module a sequential pattern <t1; . . . ; tr> is an ordered list of terms. A sequence s1 <x1; . . . ; xi> is a subsequence of another sequence s2 <y1; . . . ; yj>. Given s1 v s2, we usually say s1 is a sub pattern of s2, and s2 is a super pattern of s1. Given a pattern an ordered term set X in document d, X' is still used to denote the covering set of X, which includes all paragraphs PS. Its absolute support is the number of occurrences of X in PS that is supa. Its relative support is the fraction of the paragraphs that contain the pattern, that is, supr. A sequential pattern X is called frequent pattern if its relative support or absolute support is greater than or equal to a minimum support. The property of closed patterns can be used to define closed sequential patterns.

3.3. D-Pattern Mining Algorithm

In this module to improve the efficiency of the pattern taxonomy mining, an algorithm, SPMining, is used to find all closed sequential patterns, which used the well-known Apriori property in order to reduce the searching space. Algorithm is used to describe the training process of finding the set of d-patterns. For every positive document, the SPMining algorithm is first called giving rise to a set of closed sequential patterns SP. The main focus of this project is the deploying process, which consists of the d-pattern discovery and term support evaluation. In Algorithm all discovered patterns in a positive document are composed into a dpattern giving rise to a set of d-patterns DP. Thereafter, term supports are calculated based on the normal forms for all terms in dpatterns. Let m be the number of terms in T, n be the number of positive documents in a training set, K be the average number of discovered patterns in a positive document, and k be the average number of terms in a discovered pattern.

3.4. Inner Pattern Evolution

In this module reshuffle is used to support 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. In order to reduce the noise, we need to track which d-patterns have been used to give rise to such an error. We call these patterns offenders of nd. An offender of nd is a d-pattern that has at least one term in nd. There are two types of offenders, a complete conflict offender which is a subset of nd; and 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. The inputs of this algorithm are a set of d-patterns DP, a training set D. The output is a composed of d-pattern. The algorithm is used to estimate the threshold for finding the noise negative documents. It revise term supports by using all noise negative documents. It also find noise documents and the corresponding offenders. Shuffling is used to update NDP according to noise documents. The task of algorithm Shuffling is to tune the support distribution of terms within a d-pattern. A different strategy is dedicated in this algorithm for each type of offender. In the algorithm Shuffling, complete conflict offenders are removed since all elements within the d-patterns are held by the negative documents indicating that they can be discarded for preventing interference from these possible "noises.".

IV. RESULTS

The concept of this paper is implemented and different results are shown below, The proposed paper is implemented in Java technology on a Pentium-IV PC with minimum 20 GB hard-disk and 1GB RAM. The propose paper's concepts shows efficient results and has been efficiently tested on different Datasets.

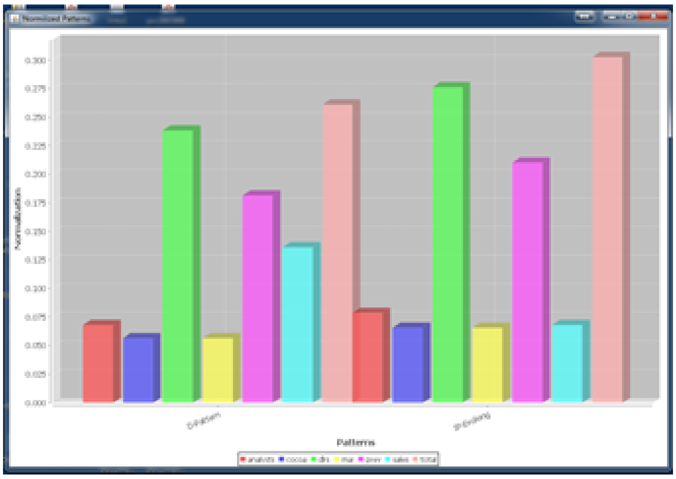

Fig. 1 Graph for Min Support of 10%.

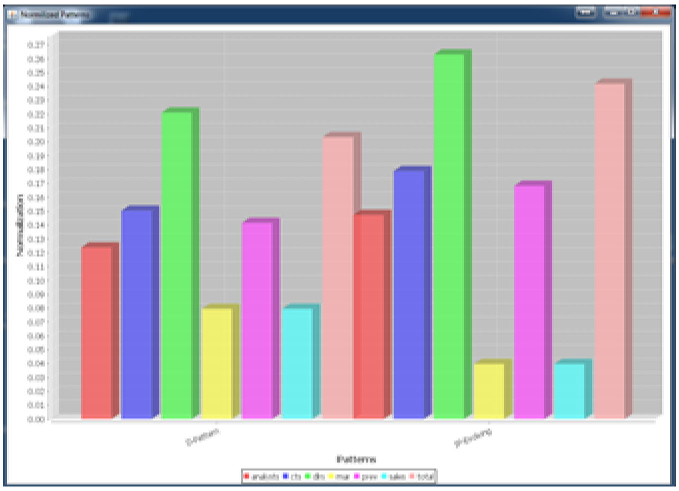

Fig. 1 Graph for min Support of 10& for large set of documents

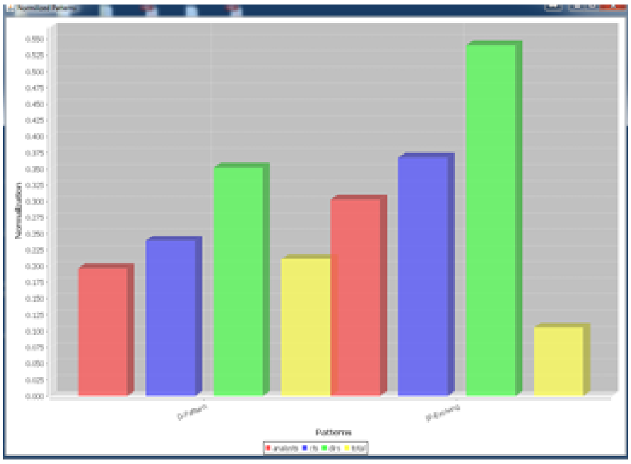

Fig. 3 Graph for min support of 15%

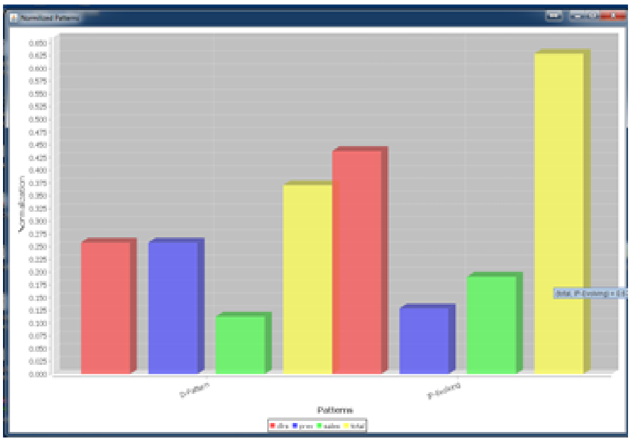

Fig. 4 Graph for min support of 15% for large set of documents

V. CONCLUSION

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). 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 problems for text mining. The proposed technique uses two processes, pattern deploying and pattern evolving, to refine the discovered patterns in text documents.

REFERENCES

[1] M. Sassano, "Virtual Examples for Text Classification with Support Vector Machines," Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP '03), pp. 208-215, 2003.

[2] 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.

[3] 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.

[4] 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.

[5] N. Cancedda, E. Gaussier, C. Goutte, and J.-M. Renders, "Word- Sequence Kernels," J. Machine Learning Research, vol. 3, pp. 1059- 1082, 2003.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] X. Li and B. Liu, "Learning to Classify Texts Using Positive and Unlabeled Data," Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI '03), pp. 587-594, 2003.

[13] 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.

14] 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.