Algarni, Abdulmohsen and Li, Yuefeng and Xu, Yue and Lau, Raymond Y.K. (2009) ***An effective model of using negative relevance feedback for information filtering***. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management, 2-6 November 2009, Asia World-Expo, Hong Kong.

An Effective Model of Using Negative Relevance Feedback for Information Filtering

Abdulmohsen Algarni, Yuefeng Li, Yue Xu
School of Information Technology, Queensland University of Technology
Brisbane, QLD, Australia
{a1.algarni,y2.li,yue.xu}@qut.edu.au

Raymond Y. K. Lau
Department of Information Systems, City University of Hong Kong
Hong Kong, China
raylau@cityu.edu.hk

ABSTRACT

Over the years, people have often held the hypothesis that negative feedback should be very useful for largely improving the performance of information filtering systems; however, we have not obtained very effective models to support this hypothesis. This paper, proposes an effective model that use negative relevance feedback based on a pattern mining approach to improve extracted features. This study focuses on two main issues of using negative relevance feedback: the selection of constructive negative examples to reduce the space of negative examples; and the revision of existing features based on the selected negative examples. The former selects some offender documents, where offender documents are negative documents that are most likely to be classified in the positive group. The later groups the extracted features into three groups: the positive specific category, general category and negative specific category to easily update the weight. An iterative algorithm is also proposed to implement this approach on RCV1 data collections, and substantial experiments show that the proposed approach achieves encouraging performance.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering

General Terms

Algorithms

Keywords

Information Filtering, Text Mining, Negative feedback, Pattern Mining

1. INTRODUCTION

A phrase (or pattern) based approach can be used to overcome the limitations of the term-based approaches and should perform better than the term-based ones. Because phrases are more discriminative and arguably carry more "semantic information". However, many studies for verifying this hypothesis were failed [5, 9, 10].

Therefore, to overcome the disadvantages of phrase-based approaches, sequential patterns were used as a promising alternative of phrases [1, 12]. The Pattern Taxonomy Model (PTM) [12] was a such model that has been proposed for IF within the data mining community and has shown encouraging improvements of effectiveness. The following comparisons drawn from the literature place PTM and IF systems in perspective: (i) PTM methods are more computationally intensive to train; (ii) sequential patterns are more effective than normal patterns; (iii) closed sequential patterns are better than frequent patterns; and (iv) too much noise in the input data (incoming document stream) adversely affects PTM systems [7].

PTM, like many filtering systems, is more reliable for using positive training documents only. One task of Relevance Feedback Trec 2008 is to satisfy the usability of using negative relevance feedback to improve filtering effectiveness. The results of the Relevance Feedback Trec 2008 indicated that using negative relevance feedback for traditional IF models did not lead to better results compared to only using positive relevance feedback (see [4] [6]).

Although, there have been several attempts to use negative feedback to improve the effectiveness of IF, negative feedback has typically been found to be far less useful than positive feedback. The existing methods of using negative feedback for IF can be categorized into two approaches. The first approach is to revise terms that appear in both positive samples and negative samples. The second approach is based on how often terms appear or do not appear in positive samples and negative samples.However, whether negative feedback can largely improve filtering accuracy is still an open question.

Based on this observation, we beliefs that using negative feedback is as important as using positive feedback to balance the extracted terms and clearly identify the boundary between positive and negative streams. This paper proposes a pattern mining based approach for using positive and negative feedback. It firstly extracts an initial list of terms from positive documents and selects some constructive negative documents (or called offenders) to reduce the space of negative feedback. It then extracts terms from negative patterns in selected negative documents. To balance the weight of extracted features, all terms are classified into three cate-

CIKM'09, November 2–6, 2009, Hong Kong, China.

gories: positive specific terms, general terms, and negative specific terms.

The remainder of this paper is organized as follows. Section 2 reviews the concepts of pattern taxonomy mining. Section 3 describes the proposed method of using negative feedback. Empirical results and discussion are reported in section 4, and the last section contains the concluding remarks.

2. PATTERN TAXONOMY MINING

We use PTM as the basic model in this study and improve it in order to use negative relevance feedback to significantly improve the performance of IF systems. For PTM, we assumed that each document d is split into a set of paragraphs $PS(d)$. 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 $T = \{t_1, t_2, \ldots, t_m\}$ be a set of terms (or keywords) that are extracted from the set of positive documents, D^+.

A sequential pattern $s =< t_1, \ldots, t_r >$ $(t_i \in T)$ is an ordered list of terms. A sequence $s_1 =< x_1, \ldots, x_i >$ is a subsequence of another sequence $s_2 =< y_1, \ldots, y_j >$, denoted by $s_1 \sqsubseteq s_2$, iff $\exists j_1, \ldots, j_y$ such that $1 \leq j_1 < j_2 \ldots < j_y \leq j$ and $x_1 = y_{j_1}, x_2 = y_{j_2}, \ldots, x_i = y_{j_y}$. Given $s_1 \sqsubseteq s_2$, we usually say s_1 is a sub-pattern of s_2, and s_2 is a super-pattern of s_1. In the following, we simply say patterns for sequential patterns.

Given a pattern (an ordered *termset*) X in document d, $\ulcorner X \urcorner$ is still used to denote the covering set of X, which includes all paragraphs $ps \in PS(d)$ such that $X \sqsubseteq ps$, i.e., $\ulcorner X \urcorner = \{ps | ps \in PS(d), X \sqsubseteq ps\}$. Its *absolute support* is the number of occurrences of X in $PS(d)$, that is $sup_a(X) = |\ulcorner X \urcorner|$. Its *relative support* is the fraction of the paragraphs that contain the pattern, that is, $sup_r(X) = \frac{|\ulcorner X \urcorner|}{|PS(d)|}$.

A sequential pattern X is called frequent pattern if its absolute support $\geq$ *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 $\exists$ any super-pattern X_1 of X such that $sup_a(X_1) = sup_a(X)$. Patterns can be structured into a taxonomy by using the *is-a* (or *subset*) relation and closed patterns.

The evaluation of term supports (weights) is different to the term-based approaches. In the term based approaches, the evaluation of a given term's weight is based on its appearance in documents. In pattern mining, terms are weighted according to their appearance in discovered patterns [11].

To improve the effectiveness of the pattern taxonomy mining, an algorithm, $SPMining(PS(d), min\_sup)$, was proposed in [12] to find all closed sequential patterns, which used the well-known *Apriori* property in order to reduce the searching space. For every positive document d, the SPMining algorithm discovered a set of closed sequential patterns based the *min_sup*.

Let SP_1, SP_2, ..., $SP_{|D^+|}$ are the sets of discovered closed sequential patterns for all documents in D^+. For a given term, its support in discovered patterns from D^+ can be described as follows:

$$support(t, D^+) = \sum_{i=1}^{|D^+|} \frac{|\{p | p \in SP_i, t \in p\}|}{\sum_{p \in SP_i} |p|}$$

Extracting patterns first then deploying them on the term space to calculate term weights would help to reduce the number of noisy terms, and give more accurate weights to terms. The obvious reason is that terms that appear in both short patterns and their super patterns would get larger weights.

3. MINING NEGATIVE FEEDBACK

Based on the document categorization system , all documents are categorized in different groups based on their topic. Each topic includes a number of levels or subtopics. This kind of categorizing can be illustrated in a topic taxonomy tree, as shown in Figure 1. To more easily organize and select the right group of a new income document, each node (topics, subtopic) in the tree is described by a number of keywords. Each child node (subtopic) can also described by the parent keywords.

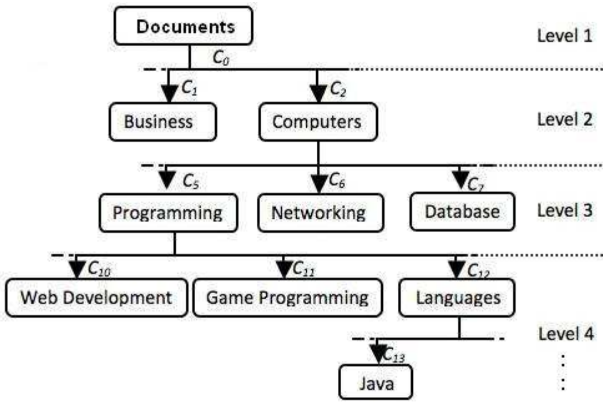

Figure 1: Documents category.

The extracted features from negative feedback documents either differ from the existing features that have been extracted from positive documents, or overlap with some existing features. Therefore, we should consider to revise weights of terms that appears in different groups. To illustrate this idea, the extracted terms are categorized into three groups based on the following definitions of *specificity* and *exhaustivity* approach:

$$exhaustivity(t) = |\{p | t \in p, p \in (DP^+ \cap DP^-)\}|$$

$$specificity^+(t) = |\{p | t \in p, p \in (DP^+ - DP^-)\}|$$

$$specificity^-(t) = |\{p | t \in p, p \in (DP^- - DP^+)\}|$$

where, DP^+ is all discovered patterns D^+, and DP^- is all discovered negative patterns of pattern taxonomies of D^-.

3.1 Strategies of Revision

This section describes the main algorithm that was generated to assemble all the previous steps for proving the theoretical ideas. We first show the basic process of revising discovered features in the training set in order to help readers understand the proposed strategies for revision.

The first step of the process is to extract initial features in all positive training documents, which includes terms and patterns, and then to select some negative samples (or offenders) in the set of negative documents in the training

***NFMining*(D)**

Input: A training set, $\{D^+, D^-\}$, parameter $\alpha = -1$; extracted features $< T, DP^+, DP^- >$, $DP^- = \emptyset$; *support* function and minimum support *min_sup*.

Output: Updated term set T and function *weight*.

Method:

1: $GT = \emptyset, T^+ = \emptyset, T^- = \emptyset$, $loop = 0$;
2: **foreach** $t \in T$ **do**
3: $\quad weight(t) = support(t, D^+)$;
4: **foreach** $d \in D^-$ **do**
5: $\quad rank(d) = \Sigma_{t \in d \cap (T \cup T^-)} weight(t)$;
6: let $D^- = \{d_0, d_1, ..., d_{|D^-|-1}\}$ in descendent ranking order, let $j = \lfloor \frac{|D^-|}{|D^+|} \rfloor$ if $loop = 0$, otherwise $j = 0$;
7: $D_3^- = \{d_i | d_i \in D^-, j \leq i < \lceil \frac{|D^+|}{3} \rceil + j\}$;
8: $DP^- = SPMining(D_3^-, min\_sup)$; //find negative patterns
9: $T_0 = \{t \in p | p \in DP^-\}$; // all terms in negative patterns
10: **foreach** $t \in (T_0 - T)$ **do**
11: $\quad$ **if** $(loop = 0)$ **then** $weight(t) = \alpha \times support(t, D_3^-)$
$\quad$ **else** $weight(t) = \alpha \times support(t, D_3^-) + weight(t)$;
12: $T^- = T^- \cup (T_0 - T)$, $loop++$;
13: **if** $loop < 3$ **then** goto step 4;
14: **foreach** $t \in T$ **do** //term partition
15: $\quad$ **if** $(t \in T^-)$ **then** $GT = GT \cup \{t\}$
$\quad$ **else** $T^+ = T^+ \cup \{t\}$;
16: **foreach** $t \in T^+$ **do**
17: $\quad weight(t) = weight(t) + weight(t) * (\frac{|\{d | d \in D^+, t \in d\}|}{|D^+|})$;
18: $T = T \cup T^-$;

set. The offender's document is selected based on the extracted features from the positive documents. Features including both terms and patterns, will be extracts from the selected negative documents using the same pattern mining technique used for feature extraction in the positive documents. In addition, this process revises the initial features and obtains revised features. The process can be repeated for several times as follows: selecting negative documents, extracting negative features and revising revised features.

Algorithm *NFMining*(D) describes the details of the strategies of the revision, where we assume that the number of negative documents is greater than the number of positive documents. For a given training set $D = \{D^+, D^-\}$, we assume that the initial features, $< T, DP^+, DP^- >$, have been extracted from positive documents D^+ before we start the algorithm, where we let $DP^- = \emptyset$. We also let the experimental parameter $\alpha = -1$ that will be used for calculating weights of terms in negative patterns.

3.2 Setting the Baseline Models

Four baseline models are used: the classic Rocchio model, a BM25 based IF model, a SVM based model, and PTM model. In this paper, our new model is called Negative Model (N-PTM).

The Rocchio algorithm has been widely adopted in the areas of text categorization and information filtering. It can be used to build the profile for representing the concept of a topic which consists of a set of relevant (positive) and irrelevant (negative) documents. The empirical parameters $\alpha = 1.0$ and $\beta = 1.0$ shows the best result in RCV1 data collection.

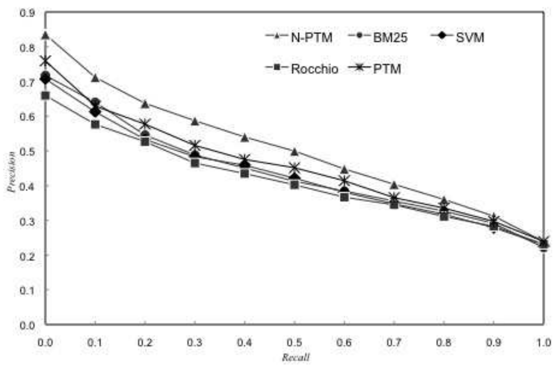

Figure 2: Comparison between used terms extracted from D^+ and from D in all assessor topics in all baseline models

Table 1: Results of PTM and N-PTM on all assessor topics.

	PTM	N-PTM	$\%chg$	P value
b/p	0.4299932	0.4684968	+8.95	0.001974031
MAP	0.4435398	0.4871910	+9.84	0.001415044
IAP	0.4641946	0.50668984	+9.15	0.002250558
$F_{\beta=1}$	0.439174956	0.4637	+5.58	0.000829231

BM25 [2, 3] is one of the other well-known term based approach that used in document retrieval. The values of the experimental parameters k_1 and b are set as 1.2 and 0.75, respectively, in this paper

Information filtering can also be regarded as a special instance of text classification [10]. SVM is a statistical method that can be used to find a hyperplane that best separates two classes. To compare with other baseline models, we tried to use SVM to rank documents rather than to make binary decisions. For this purpose, threshold b can be ignored [7].

PTM model is also selected as one of the baselines models because we want to verify that the negative relevance feedback are important as the same as positive feedback in the topic filtering. In PTM model we set the minimum support, $min\_sup = 0.2$, and the size of the term set is 4000.

To evaluate the result the Reuters Corpus Volume 1 (RCV1) [8] was used to test the effectiveness of the proposed model. In this paper Precision (p) and Recall(r) are suitable because the measure how precise and how complete the classification is on the positive class. The F-score (also called the F_1-score) is often used to compare classifiers in the IF area. The N-PTM model is compared with PTM, Rocchio, BM25, and SVM models for each variable b/p (breakeven point), MAP (average precision), IAP (Interpolated Average precision), $F_{\beta=1}$ over all assessor topics, respectively.

4. DISCUSSION

Table 1, 2 and Figure 2, shows the results in all assessor topics for the baseline models and proposed method, where N-PTM is the proposed method in this paper. The experiment results clearly indicate that the proposed method using both positive and negative training documents achieve an improved result.

Table 2: Results of assessor topics where $\%chg$ is the percentage change over the best term-based model.

	Rocchio	BM25	SVM	N-PTM	$\%chg$
b/p	0.420	0.403	0.409	0.468	11.52
MAP	0.430	0.417	0.409	0.487	13.17
IAP	0.452	0.439	0.434	0.507	12.03
$F_{\beta=1}$	0.430	0.421	0.421	0.464	7.88

Generally, negative is a term that can be defined as anything except something positive. It is obvious that not all negative feedback is suitable to be selected as an offender, where offenders are the most useful negative documents that can help balance the weight of general terms because they are closer to the user's interested subtopic. Figure 3 shows the difference between using all negative documents and using offenders for all the assessor topics. The proposed method for offender selection is shown to meet the design objectives.

To review the weight of extracted features, the proposed method classifies extracted terms into general terms and specific terms, which is a distinct advantage compared with other methods. Specific terms are generally considered to be more interesting than general terms for a given topic. However, general terms are still important because they frequently appear in positive documents. The problem for general terms is that they may also frequently appear in some negative documents, probably because negative documents describe some extent to what users need. Before revision in the top 10 topic more than 72% weights are distributed to general terms, although the percentage of general terms is 31% for all extracted terms in the positive documents.

To reduce the side effects of using general terms in the extracted features, the proposed method adds negative specific terms into the extracted features. However, adding negative specific terms to balance the negative documents will also affect the positive document because they share the same general terms. For this problem, the proposed method only increases the weights of positive specific terms when it conducts the revision using negative documents. After revision, the percentage of general terms weight drooped into 59%, because about 175 negative specific terms added in each topic. Figure 3 shows the proposed model in different stages. Compared with the best state-of-the-art models, the proposed approach achieves excellent performance with 11.08% average change for all four measures.

5. CONCLUSIONS

Negative feedback contains information that helps to improve feature selection and balance the extracted term weights. However, one of the common problems for negative feedback is that negative has no clear defined boundary. As a result, it is important to carefully select offender documents in order to reduce the space of negative documents. In this paper, we proposed a new approach to use both positive and negative feedback to improve PTM effectiveness. The results compared with several baseline models, including Rocchio, SVM, and BM25. The experimental results on RCV1 collections and TREC topics shows that the proposed method achieves exciting performance with 11.08% average percentage change for all four measures. This research would be a significant contribution to information filtering for using negative relevance feedback.

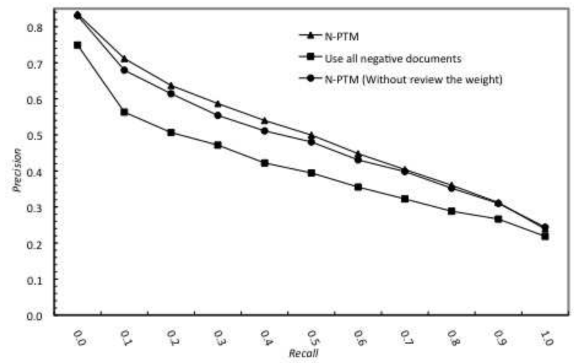

Figure 3: Comparison of the different stages of proposed method.

6. REFERENCES

[1] N. Jindal and B. Liu. Identifying comparative sentences in text documents. In *SIGIR*, pages 244–251, 2006.

[2] K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - part 1. *Inf. Process. Manage.*, 36(6):779–808, 2000.

[3] K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - part 2. *Inf. Process. Manage.*, 36(6):809–840, 2000.

[4] R. Kaptein, J. Kamps, and D. Hiemstra. The impact of positive, negative and topical relevance feedback. In *TREC*, 2008.

[5] D. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *SIGIR*, pages 37–50, 1992.

[6] B. Li, F. Liu, and Y. Liu. Utdallas at trec 2008 blog track. In *TREC*, 2008.

[7] Y. Li, X. Zhou, P. Bruza, Y. Xu, and R. Y. Lau. A two-stage text mining model for information filtering. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1023–1032, Napa Valley, California, USA, 2008.

[8] T. Rose, M. Stevenson, and M. Whitehead. The reuters corpus volume 1 - from yesterdays news to tomorrows language resources. In *In Proceedings of the Third International Conference on Language Resources and Evaluation*, pages 29–31, 2002.

[9] S. Scott and S. Matwin. Feature engineering for text classification. In *The 16th International Conference on Machine Learning*, pages 379–388, 1999. Scott, Sam and Matwin, Stan.

[10] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.

[11] S.-T. Wu. *Knowledge discovery using pattern taxonomy model in text mining.* PhD thesis, Queensland University of Technology, 2007.

[12] S.-T. Wu, Y. Li, and Y. Xu. Deploying approaches for pattern refinement in text mining. In *ICDM*, pages 1157–1161, 2006.