
© Copyright 2009 ACM
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 uses 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 such a 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 believe 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-
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 sub-sequence of another sequence \( s_2 = < y_1, \ldots, y_j > \), denoted by \( s_1 \subseteq s_2 \), iff \( x_1 \leq y_1 < \cdots < x_i \leq y_j \) such that \( 1 \leq j_1 < j_2 < \cdots < j_2 \leq j \) and \( x_1 = y_{j_1}, x_2 = y_{j_2}, \ldots, x_i = y_{j_i} \). Given \( s_1 \subseteq s_2 \), 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 \( d \), \( \sup(X) \) is still used to denote the covering set of \( X \), which includes all paragraphs \( \text{ps} \in PS(d) \) such that \( X \subseteq \text{ps} \), i.e., \( \sup(X) = \{ \text{ps} | \text{ps} \in PS(d), X \subseteq \text{ps} \} \). Its absolute support is the number of occurrences of \( X \) in \( PS(d) \), that is \( \sup(X) = |\sup(X)| \). Its relative support is the fraction of the paragraphs that contain the pattern, that is, \( \sup(X) = \frac{|\sup(X)|}{|PS(d)|} \).

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 \( \sup(X_1) = \sup(X) \). 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, \( \text{SPMining}(PS(d), \min_{\sup}) \), 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 \( \text{SPMining} \) algorithm discovered a set of closed sequential patterns based the \( \min_{\sup} \).

Let \( SP_1, SP_2, \ldots, SP_{D^+} \) be 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:

\[
\text{support}(t, D^+) = \sum_{t_i} \frac{|\{ p | p \in SP_i, t \in p \}|}{\sum_{p \in SP_i} |p|}
\]

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 be described by the parent keywords.

![Figure 1: Documents category.](image-url)

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:

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

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

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

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 \(\text{min}\_\text{sup}\).

**Output:** Updated term set \(T\) and function weight.

**Method:**
1: \(GT = \emptyset, T^+ = \emptyset, T^- = \emptyset, \text{loop} = 0\);
2: foreach \(t \in T\) do
3: \(\text{weight}(t) = \text{support}(t, D^+)\);
4: foreach \(d \in D^-\) do
5: \(\text{rank}(d) = \sum_{t \in d \cap (T^+ \cup T^-)} \text{weight}(t)\);
6: let \(D^- = \{d_0, d_1, ..., d_{|D^-| - 1}\}\) in descendent ranking order,
   \(\text{let } j = \lceil |D^+| + 3 \rceil\) if loop = 0, otherwise \(j = 0\);
7: \(D^-_j = \{d_0|d_j \in D^- \cup \{t_j \in \lceil |D^+| + 1 \rceil\}\};\)
8: \(DP^- = \text{SPM}(D^-_0, \text{min}\_\text{sup}); //\text{find negative patterns}\)
9: \(T_0 = \{t \in D^- \cap D^-\}; //\text{all terms in negative patterns}\)
10: foreach \(t \in (T_0 \setminus T)\) do
11: \(\text{if (loop} = 0)\) then \(\text{weight}(t) = \alpha \times \text{support}(t, D^-_0\)
   else \(\text{weight}(t) = \alpha \times \text{support}(t, D^-_0) + \text{weight}(t)\);
12: \(T^- = T^- \cup (T_0 \setminus T), \text{loop} + +\);
13: \(\text{if } \text{loop} < 3 \text{ then goto step } 4\);
14: foreach \(t \in T\) do \(/\text{term partition}\)
15: \(\text{if } (t \in T^-)\) then \(GT = GT \cup \{t\}\)
   else \(T^+ = T^+ \cup \{t\}\);
16: foreach \(t \in T^+\) do
17: \(\text{weight}(t) = \text{weight}(t) + \text{weight}(t) * (\frac{|\{d \in \emptyset \cap (D^+_t \cup D^-_t)\}|}{|D^+_t|})\);
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 extracted 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.

<table>
<thead>
<tr>
<th></th>
<th>PTM</th>
<th>N-PTM</th>
<th>%chg</th>
<th>(P) value</th>
</tr>
</thead>
<tbody>
<tr>
<td>b/p</td>
<td>0.4299932</td>
<td>0.4684908</td>
<td>+9.84</td>
<td>0.001974031</td>
</tr>
<tr>
<td>MAP</td>
<td>0.4435398</td>
<td>0.4871910</td>
<td>+9.15</td>
<td>0.001415044</td>
</tr>
<tr>
<td>IAP</td>
<td>0.4641946</td>
<td>0.5066898</td>
<td>+9.15</td>
<td>0.002250558</td>
</tr>
<tr>
<td>(F_{\beta=0.5})</td>
<td>0.439174956</td>
<td>0.4637456</td>
<td>+5.58</td>
<td>0.00429223</td>
</tr>
</tbody>
</table>

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, \(\text{min}\_\text{sup} = 0.2\), 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), \(F_{\beta=0.5}\) 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 experimental 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 \( %\text{chg} \) is the percentage change over the best term-based model.

<table>
<thead>
<tr>
<th></th>
<th>Rocchio</th>
<th>BM25</th>
<th>SVM</th>
<th>N-PTM</th>
<th>( %\text{chg} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/p</td>
<td>0.420</td>
<td>0.403</td>
<td>0.409</td>
<td>0.468</td>
<td>11.52</td>
</tr>
<tr>
<td>MAP</td>
<td>0.430</td>
<td>0.417</td>
<td>0.409</td>
<td>0.487</td>
<td>13.17</td>
</tr>
<tr>
<td>IAP</td>
<td>0.452</td>
<td>0.439</td>
<td>0.434</td>
<td>0.507</td>
<td>12.03</td>
</tr>
<tr>
<td>( F_{\beta=1} )</td>
<td>0.430</td>
<td>0.421</td>
<td>0.421</td>
<td>0.464</td>
<td>7.88</td>
</tr>
</tbody>
</table>

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 show that the proposed method achieves exciting performance with 11.08% average change for all four measures. This research would be a significant contribution to information filtering for using negative relevance feedback.

6. REFERENCES