Guided Navigation Using Query Log Mining through Query Expansion

Burcu Yurekli, Gokhan Capan, Baris Yilmazel, Ozgur Yilmazel
Computer Engineering Department
Anadolu University
Eskisehir, Turkey
{byurekli, gcapan, ibyilmazel, oyilmazel}@anadolu.edu.tr

Abstract—We introduce a method to help people to refine their search queries based on other people’s similar queries. We compared two different methods of collaborative filtering on search logs of a popular e-commerce site to do query expansion. In item-based method, by identifying similarities between different items, and in user-based method, by identifying similarities between different users, we compute recommendations for users. The experimental results show that item-based query expansion method provides better performance than the user-based method. While the user-based method improved the system performance 20%, the item-based method provided success of 73%. Also, item-based method recommended better expansion terms than the user-based method, which is important in helping web users to easily access information needs by formulating qualified queries. This document is a comparison of these two methods, and their effects on system performance.

Keywords-query expansion; query log mining; guided navigation; performance

I. INTRODUCTION

As Internet has become widely accessible, the amount of available online data and the number of users on web sites have grown enormously. In recent years, this excessive information increase and the lack of organized information resources on World Wide Web, make it very difficult for the web users to reach the information they need without being overwhelmed. The users need to formulate highly-qualified queries with appropriate words to be able to answer their information needs. However, formulating high-qualified queries is a difficult task for the web users. Observations show that generally web users present short queries that have less than two words to search engines [1].

To help web users to formulate better queries, researchers have focused on query expansion methods [2]. In query expansion, by suggesting additional query terms, users give additional input on query words or phrases. Some web search engines, like Yahoo and Google, suggest related queries or related searches in response to a query to their users and the users have a chance to use one of these alternative queries to refine their search [3].

In this paper, we implemented two different collaborative filtering algorithms to suggest new query terms or phrases to users, based on the analysis of user logs. Our aim is to help users to reformulate their queries by recommending new alternative queries.

II. BACKGROUND

Existing query expansion approaches can be classified as either global or local. Approaches based on global analysis, aims to obtain expansion terms by discovering corpus-wide term relations based on co-occurrence analysis on the whole corpus. Term clustering [4] is one of the earliest global analysis techniques that groups terms into clusters according to their co-occurrences. Queries are then expanded by using these clusters. Other well-known global analysis techniques include Latent Semantic Indexing [5], similarity thesauri [6] and PhraseFinder [7]. In global analysis techniques, to expand a query, a similarity matrix among terms needs to be constructed and this requires the calculation of co-occurrence statistics for every term pair. Even though these techniques are robust, they demand considerable amount of computing resources [2].

The local query expansion techniques uses only a set of documents retrieved for the query, thus they are more focused on the given query than global analysis [8]. The simplest local analysis technique is relevance feedback [9], [10], which is based on user feedback information. However, to achieve good performance, sufficient relevance judgments are required and that makes the use of the technique difficult. Due to the lack of sufficient relevance judgments, techniques based on local feedback are developed that uses well-known relevance feedback procedure used in IR, for query expansion. The chosen terms for query expansion with local feedback techniques are the most frequent terms from the top ranked documents [11], [12]. The drawback of this technique is that if most of the top ranked documents are irrelevant, then the expansion terms are unrelated to the given query. In recent years many researches have proposed techniques to improve local feedback. [13] proposed a query expansion technique called local context analysis that combines both global and local techniques.

III. DATA SET

In this study, we used the query logs of one of the most popular e-commerce sites in Turkey. These query logs were gathered by session id and search text. From the query logs we extracted queries of a user within a session. Among these
obtained is 67761*119634, but there are only 293100 entries, by 67761 different people (users). The session-query matrix had 119634 different log queries (items) that were searched at least 3, and at most 30 queries.

We chose the most popular, moderate and rare 10 queries randomly. We grouped the queries according to search frequencies. The queries which were searched more than or equal to 1000 times grouped as popular queries, which were searched between 200 and 500 times grouped as moderate queries, and which were searched less than or equal to 50 times grouped as rare queries. From each of these groups, randomly 10 queries were chosen. Fig. 1 lists the 30 queries used in the experiments.

### IV. Experiments

We implemented two different collaborative filtering algorithms to find related query terms for the 30 queries given in Fig. 1.

#### A. User-Based Method (Session Based)

We used Apache Mahout’s Taste module, which is a flexible, fast collaborative filtering engine for Java [14]. We applied Mahout’s User-Based Recommender demo on the data set. The sessions were taken as the users, and the queries were taken as the items.

We used Boolean Tanimoto Coefficient Similarity as the user correlation algorithm [15]. Boolean Tanimoto is a variant of Tanimoto Coefficient Similarity, in which implementation of similarity is based on Tanimoto Coefficient or extended Jaccard Coefficient. Since our data contains 0s and 1s we chose Boolean Tanimoto that is appropriate for use with boolean classes.

Then we applied Mahout’s Nearest N User Neighborhood algorithm that computes a neighborhood consisting of the nearest n users to a given user. For each session, we obtained the most similar n sessions, where the nearest n sessions (users) defined by Boolean Tanimoto Coefficient Similarity.

From these results we obtained the query recommendations for the randomly chosen 30 queries. This method is one of the most known collaborative filtering methods.

#### B. Item-Based Method (Query Based)

In contrast to User-Based Method, Item-Based method is based on query similarity, not on user similarity. The idea is to recognize relations between items by analyzing the user-item matrix and for a given pair predicate related items based on these relations. In other words, this method first computes similarity between items and then selects the most similar ones [16]. In determination of similarities, we used Log-likelihood ratio [17], [18], which is a statistical analysis for computing similarity. We estimate that if two items are similar to each other, than they should be searched together.

Algorithm:

```
Query q;
For each query q {
    Compute similarity between each q and query
}
```

We used the same query log data set as in the item-based method. For each query we computed similarity between that query and all other queries with Log-likelihood ratio. Based on these similarities, 50 queries that are most similar to that query have chosen. Among the set of most similar queries, the ones which had more than or equal to 0.7 rank assigned as similar and added to similarity map. Fig. 3 shows the construction of the similarity map for item-based collaborative filtering.

When a new query is submitted, by using this similarity map, we can get the related queries recommended for this query. By this way, for each of the 30 queries in the experimental query set we obtained query recommendations. Fig. 4 illustrates the top 50 expansion terms recommended for the query “iphone” by item-based method.
Figure 3. Construction of the similarity map for item-based collaborative filtering method.

Figure 4. Expansion terms for “iphone” with item-based method.

Figure 5. Relevancy comparison for the results of user-based method.

Figure 6. Relevancy comparison for the results of item-based method.

Table 1 shows the number of relevant expansion terms recommended by the user-based and item-based collaborative filtering methods. As can be seen from Fig. 5, Fig. 6, and the results of Table 1, item-based method recommends better expansion terms than the user-based method. The comparison of the top 50 expansion terms produced for the query “iphone” by both methods also shows this. “touch”, “anycool”, “lg cookie”, “iphone 16gb”, “blackberry”, “htc” “iphone kulaklık (iphone headphones)”, “iphone kılıf (iphone case)” and “dokunmatik telefonlar (touch screen phones)” are some of the good terms recommended by the item-based method that can be very useful for the users of an e-commerce site. On the other hand, when we analyzed the recommendations of the user-based method, between the results of all 30 queries we observed “fantazi gece ayakkabileri (fantasy evening shoes)” recommendation. The feature of this query is that it has been asked by one user, only one times, and in that session the user looked for three items.

V. EXPERIMENTAL RESULTS

The top 50 expansion terms for all the 30 queries found with user-based and item-based methods were judged manually by 5 human assessors. Each assessor judged the result either as 1 or 0, meaning relevant or not relevant, respectively.

To assess how consistent the ratings of the judges, we calculated the average pairwise kappa [19]. For the user-based method, the kappa measure is calculated as 0.7966 and for the item-based method kappa measure is found as 0.8072. This kappa values show that there is an agreement between these judges.

A. Quality of Expansion Terms

For all the 30 queries in the experimental query set, we analyzed the quality of the expansion terms found by the system. For each of the 50 expansion terms of a query, if at least three assessors agree that the expansion term is related to the query, then that term is assigned as relevant. Fig. 5 and Fig. 6 show the number of relevant and non-relevant expansion terms appointed for each query by the assessors based on the results of user-based and item-based methods, respectively.

### Table 1. Comparison of the Two Collaborative Filtering Methods on Relevant Percentage of Expansion Terms

<table>
<thead>
<tr>
<th>Relevant Terms (%)</th>
<th>User-Based Method</th>
<th>Item-Based Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>26.32</td>
<td>62.20</td>
<td></td>
</tr>
</tbody>
</table>
B. Performance Comparison of the Systems

The overall performance of the systems is derived by considering the number of relevant queries. For each query in the experimental query set, the average score that is given by an assessor for 50 expansion terms has been calculated. Among the five average scores given by the assessors, the highest one has chosen as the score of the query. Queries with score greater than or equal to 0.7 assigned as relevant.

<table>
<thead>
<tr>
<th>Relevant Queries (%)</th>
<th>User-Based Method</th>
<th>Item-Based Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevant Queries (%)</td>
<td>20</td>
<td>73.33</td>
</tr>
</tbody>
</table>

Overall performances of the systems for both user-based and item-based methods are shown in Table 2. Among all 30 queries in the experimental set, user-based method recommended 6 of them correctly and it improved the performance of the system 20%. The item-based method recommended 22 queries correctly and by using item-based method, performance of the system improved 73.33%.

The performance of the user-based method also examined for different neighborhood sizes, such as 1, 5, 10, 50 and 100. However, the neighborhood size did not much affect the performance of the user-based method, even for small neighborhood sizes there were unrelated expansion terms.

VI. CONCLUSIONS & FUTURE WORK

For web users to easily access the information they need, we developed two different collaborative filtering methods to recommend query expansion terms based on query logs. For our experiments we used the query logs of a popular e-commerce site. This experiments show that the success of the item-based method, both in system performance and quality of expansion terms, is better than the user-based method in finding similar terms. While the success of the user-based method is 20%, the item-based method provided success of 73%.

Item-based method can achieve better results, but in run time user-based method is faster. To make the item-based method faster, it was considered as batch, and pre-computed similarity values were used in computations. The idea is that the similarity map does not have to be created each time, instead it will be created once, and than according to user queries this map will be updated periodically. By this way, item-based method achieves very high on-line performance, especially for large data sets.

When the item-based method begins to be used in a system, such as an e-commerce web site, according to the user selections the performance of the system will be further increased. People using this system will choose the relevant recommendations of a query from the suggested ones, so after an update when the new similarity map is created, the recommendations that were not chosen as relevant by the users will not enter into new similarity map.

This system provides lots of benefits for both web users and e-commerce businesses. For the web users;
- The system can be used for query expansion. According to the suggested recommendations users can refresh their queries.
- This system makes it easy for the users to reach their information needs. Users are given a chance to make choices on the suggested recommendations. For example, the user who asked the query “fan”, may be looking for “işlemci fanı (cpu fan)”, “vantilatör (ventilator)”, “pervane (propeller), or “laptop soğutucu (laptop cooler)”. Showing relevant search queries in this way will provide easier access to user information needs.

On the other hand, this system can help e-commerce business by supplying more sales.

In our future work, we plan to include the clickstream data to our work. For each session in the query logs, in addition to search text, we will keep the set of documents which the user clicked on. By this way, query expansion will be performed not just by looking at the queries, but also analyzing the related documents in the logs.

REFERENCES


