Adaptive Clustering of Search Results in Interactive Information Retrieval

Xuehua Shen[1], ChengXiang Zhai[2], and Nicholas J. Belkin[3]

[1] Google Incorporation
[2] Department of Computer Science, University of Illinois at Urbana-Champaign
[3] School of Communication, Information and Library Studies, Rutgers University

Abstract. Clustering of search results has been shown to be advantageous over the simple list presentation of search results. However, in most clustering interfaces, the clusters are not adaptive to a user's interaction with the clustering results, and the important question "how to optimize the benefit of a clustering interface for a user" has not been well addressed in the previous work. In this paper, we study how to exploit a user's clickthrough information, which is naturally available when a user is interacting with a clustering interface, to adaptively reorganize the clustering results and help a user find the relevant information more quickly. We propose four strategies for adapting clustering results based on user actions, including reranking documents based on a selected cluster, reranking documents based on a viewed document, merging unselected clusters, and promoting "near-miss" documents. Evaluation of the utility of a cluster presentation of results is a challenging task. We propose a general method to simulate different kinds of users and linearize the cluster results so that we can compute regular retrieval measures. The simulation experiments show that the adaptation strategies have different performance for different types of users; in particular, they are effective for "smart users" who can correctly recognize the best clusters, but not effective for "dummy users" who follow system's ranking of results. We further conduct a user study on one of the four adaptive clustering strategies to see if an adaptive clustering system using such a strategy can bring users better search utility and/or experience than a static clustering system. The results show that there is generally no significant difference between the two systems from a user's perspective.

1 Introduction

The main goal of a search engine is to rank relevant documents above non-relevant ones. There has been a lot of research in developing effective retrieval models to help achieve this goal. For example, many effective retrieval models have been developed, including models for content-based ranking, such as the vector-space model [12], classic probabilistic models [9], language models [17], and algorithms for link-based ranking, such as PageRank [8].

However, an equally important goal of a search engine is to present the search results effectively so that a user can find the relevant information from the results quickly. Compared with the huge amount of literature on retrieval models, there is little research on how to optimize result presentation for a user.

Indeed, most search engines present a ranked list of documents with brief summaries. In an ideal case when the system performs extremely well on a query, a user would be able to find most or all the relevant documents on the top of this list, and such a presentation would be fine. However, due to the inevitable mismatches between a query and documents, the search results are more often non-optimal. Indeed, it is quite common that a user may not find any relevant document among the top ranked ones. In such a case, the user would have to go through the many non-relevant documents in the list until eventually finding some relevant ones. Intuitively, a clustering view of the search results would be much more useful in such a case. Indeed, clustering of search results has been shown to be an effective way to present the search results [16, 3, 4, 1], and has been adopted by some search engines such as vivisimo [4] and carrot2 [5].

Although clustering of search results has been studied, in most existing work, the clusters are generally not adaptive to a user's interaction with the clustering results. In Scatter/Gather[2, 5], the authors proposed that re-clustering can be performed on the user-selected clusters (i.e., "scattering"), which can be regarded as attempting to adapt the clustering results to a user based on the user's action. Unfortunately, this is as far as the work goes, and to the best of our knowledge, there has been no work that attempts to seriously study the important question "how to optimize the benefit of a clustering interface for a user."

With non-adaptive clustered results, a user would generally select a cluster and examine the results in it. Thus the utility of the results is largely determined by the clustering algorithm. Intuitively, however, the clustering results may be improved as the system sees more user interactions when the user examines the results. For example, when a user opens a particular cluster for viewing, we can infer from this action that the user likes the content of this cluster better than others, and this knowledge can be exploited immediately to rerank the results inside the cluster to be viewed as well as the results in other clusters. Also, since the unselected clusters may not be so interesting to the user, we may consider merging all of them into one cluster. Moreover, since we now know the user likes the selected cluster better than other clusters, we could move some "borderline" documents that were put in other clusters into the selected cluster. Such adaptation can be expected to improve the utility of the clustered results in the sense that it would help a user find relevant information more quickly.

In this paper, we study how to exploit a user's clickthrough information, which is naturally available when a user is interacting with a clustering interface, to adaptively reorganize the clustering results and help a user find the relevant information more quickly. Specifically, we propose four strategies for adapting clustering results based on user actions, which are (1) reranking documents based on a selected cluster, (2) reranking documents based on a viewed document, (3) merging unselected clusters, and (4) promoting "near-miss" documents. Evaluation of the utility of a cluster presentation of results is a challenging task. We propose a general method to simulate different kinds of users and linearize the cluster results so that we can compute regular retrieval measures. The method can be used to compare different clustering results as well as comparing clustering results with a regular ranked list of results. The simulation experiments show

[4] http://vivisimo.com/
[5] http://www.carrot2.org/

that the adaptation strategies have different performance for different types of users; in particular, they are effective for "smart users" who can correctly recognize the best clusters, but not effective for "dummy users" who simply follow system's ranking of results. Among the four proposed adaptation strategies, the strategy of reranking based on viewed document is shown to be most effective, but other strategies are also beneficial. We further conduct a user study to see if an adaptive clustering system can bring improved search utility and/or experience to users, compared with a static clustering system. To ensure that we have enough users to make meaningful conclusions, we focus on the strategy of promoting near-miss documents. We chose this strategy because it has been shown to be effective in the simulation study and also affects the clustering membership so that it is likely to make users feel some difference. We recruited 24 subjects and did a study of comparing a static clustering baseline system with an adaptive clustering system in Web search. The results show that there is generally no significant difference between the two systems from a user's perspective. Specifically, more users say that they like the adaptive system better but users saved more relevant documents with the static interface than with the adaptive interface. While this result is a bit disappointing, it is also not very surprising because in general, a user tends not to feel much difference unless there is a significant difference in the interface. Overall, our study shows that adaptive clustering has a good potential for improving search utility for users, but a user may not perceive any significant difference in the system.

The rest of the paper is organized as follows. We first discuss related work in Section 2. We then describe the basic retrieval method, clustering algorithm, and query updating method in Section 3. We propose four adaptive clustering strategies in Section 4, and present a simulation study of the four strategies in Sections 5. In Sections 6, we present our user study. Finally, we conclude in Section 7.

2 Related Work

Clustering of search results has been extensively studied. In [2, 5], Scatter/Gather is proposed and evaluated as a document browsing method. In the evaluation, they consider the cluster with the largest number of relevant documents as the best cluster and linearize the best cluster according to the rank of documents in the cluster. This short linearized ranked list is compared with the equivalent number of documents in the original ranked list. In [3], the effectiveness of grouping search results is studied and it is found that presenting search results in context (with category information) is more effective than simply presenting search results as a ranked list.

There are some studies of clustering web search results. In [16], a web search clustering algorithm is applied to a search engine for clustering search results. This work shows the feasibility of clustering web search results on-the-fly. In [4], the same clustering interface for web search is implemented using a hierarchical clustering algorithm. Vivisimo is a meta-search engine doing search result clustering.

Our work differs from the previous work on clustering in that instead of presenting a static clustering results as done in most previous work, our clustering results will dynamically change during the user's interaction with the search engine.

Our work is also related to some recent work on implicit feedback (e.g., [13]) where clickthrough information is used to rerank documents. The difference is that we exploit implicit feedback information collected while a user is browsing clustering results and we attempt to reorganize clustering results while the previous work is based on a simple list presentation of search results.

3 Basic Algorithm Description

Information Retrieval Method We use the vector space model with Okapi BM25 term frequency weighting as our baseline retrieval method [10]. Specifically, the query vector is

$$\boldsymbol{q} = (q_1, ..., q_{|V|})$$

where q_i is the weight for term w_i in our vocabulary V, and is given by the following TF-IDF weighting formula:

$$q_i = \frac{(k_3+1)c(w_i,q)}{k_3 + c(w_i,q)} \log \frac{N+1}{df(w_i)+0.5}$$

where $c(w_i, q)$ is the count of word w_i in the query q, N is the total number of documents in the collection, $df(w_i)$ is the number of documents that contain the term w_i, and k_3 is a parameter.

The document vector is

$$\boldsymbol{d} = (d_1, ..., d_{|V|})$$

where d_i is the weight for term w_i in our vocabulary V, and is given by the following TF-IDF weighting formula:

$$d_i = \frac{k_1 c(w_i,d)}{c(w_i,d) + k_1(1 - b + b \cdot \frac{dl}{avdl})} \log \frac{N+1}{df(w_i)+0.5}$$

where $c(w_i, d)$ is the count of word w_i in the document d, dl is the document length, $avdl$ is the average document length, and k_1 and b are parameters.

The score of document d w.r.t. query q is computed as the dot product of $\boldsymbol{q}$ and $\boldsymbol{d}$. k_1, b and k_3 are parameters and set to 1.2, 0.75 and 1000, as recommended in [10].

K-Medoid Clustering After the retrieval system retrieves a ranked list of results using the baseline retrieval method, the top N ranked results will be clustered into K clusters using the K-Medoids (a.k.a. PAM) algorithm, which is a partition based clustering algorithm [7]. A "Medoid" is the most centrally located object in a cluster. K-Medoids algorithm starts with an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it reduces the total distance of the resulting cluster. K-Medoids algorithm is more robust than K-Means in presence of noise and outliers because a medoid is less influenced by outliers than a mean. The similarity of two documents is computed as the dot product of two corresponding document vectors as defined above.

Query Updating (Adapation) Given a selected cluster C by the user, we can update the query using all or a subset of documents in the selected cluster using a feedback method. When the user selects a document to view, we could also update the query vector based on just one single document for feedback. In our study, we use a modified Rocchio feedback method [11], which is defined as:

$$\overrightarrow{q}' = \overrightarrow{q} + \alpha \frac{1}{M} \sum_{d \in C} \overrightarrow{d} \tag{1}$$

where α is the interpolation coefficient. This coefficient can be chosen by running experiments on another set of query topics with known relevant documents in the collection. M controls how many documents we select from the cluster to update the query.

We set α to 0.5 in all our experiments based on tuning on some training queries as described in Section 5. We set M to 6 in most experiments of this work. We also control the number of terms used for query updating by setting it to 20.

4 Four Adaptive Strategies

We propose four strategies for adapting clustering results based on user actions, including reranking documents based on a selected cluster, reranking documents based on a viewed document, merging unselected clusters, and promoting "near-miss" documents, which we will discuss below.

Reranking Based on Cluster Selection When a user selects a cluster to view, we may infer that the user likes the selected cluster better than un-selected cluster(s). This information can be exploited to improve the ranking of documents within a cluster. The information about the selected cluster can be combined with the original query to rerank documents in other (unselected) clusters. Here the retrieval system does not change the cluster structure, i.e., the retrieval system does not move one document from one cluster to another cluster. Instead, documents within each cluster are reranked according to dot product scores of the updated query vector and document vectors. The updated query is computed according to Equation 1.

Reranking Based on Document Selection When a user clicks on a document to view after selecting a cluster, the viewed document can presumably provide more information about what the user is interested in to the retrieval system and can thus be exploited to improve search results (i.e., implicit feedback [13]). In this strategy, the retrieval system would use the viewed document or snippet to update the query as:

$$\overrightarrow{q}' = \overrightarrow{q} + \alpha \overrightarrow{d} \tag{2}$$

where $\overrightarrow{d}$ is the selected document vector. The updated query can be used to rerank the documents within each cluster. Here, the cluster structure does not change either.

Merging Unselected Clusters When the user selects a cluster or views a document, the adaptive clustering algorithm can also restructure the clusters. In [2, 5], the retrieval system merges several relevant clusters according to a user's selection. After seeing a user selecting a cluster, it would be reasonable to assume that the user may not be so interested in the partitioning of search results in other clusters. Thus, the retrieval system can merge all unselected clusters into a big cluster and put this big cluster below the selected cluster. Then according to the updated query as computed by Equation 1, the retrieval system can rerank all the documents in the big cluster. In this strategy, the cluster structure is changed.

Promoting "Near Miss" Documents In this strategy, when the user clicks on a cluster, the adaptive clustering algorithm presents not only the documents in the clicked cluster to the user, but also some (borderline) documents which were originally scattered into unselected clusters. Specifically, the retrieval system would select those documents from each unselected cluster that are most similar to the updated query vector, and then insert these "near miss" documents into the bottom of the selected cluster. The updated query is computed using Equation 1. Using this strategy, the cluster structure is also changed.

5 Simulation Study

5.1 Experiment Methodology

Following the use of the simulation strategy in some previous work [5, 15], we also use the simulation strategy to evaluate the proposed adapatation strategies.

Different from previous work (e.g., [5]), our simulation distinguishes two different types of extreme users. One is "smart users", who can always make intelligent decisions. Such a user would be assumed to always select the best cluster among several clusters according to the description of the cluster labels and alwasy select a relevant document among a set of documents to view according to the snippet of each document. We simulate the interaction of the smart user as follows. We compute the percentage of relevant documents in each of the K clusters, sort the K clusters according to the percentage of relevant documents in a cluster. We assume the smart user will select the top (also the best) cluster. We linearize or expand these K clusters into a ranked list and evaluate the expanded ranked list with regular retrieval performance measures. We refer to the retrieval performance of this ranked list as the *smart baseline*. After a smart user selects the cluster with the highest percentage of relevant documents, we consider this user interaction as offering an opportunity for implicit feedback. The retrieval system can thus update the user's information need and rerank documents in each cluster. We evaluate the performance of the new ranked list and refer to the result as the *smart adaptive*. After the documents of the best cluster are presented, the smart user would select the best document in the best cluster to view. We simulate this behavior by selecting the top ranked *relevant* document in the best cluster. Immediately after the smart user selects the best document in the best cluster to view, the retrieval system will further update the user's information need and rerank documents in each cluster again. We refer to this result as the *smart reranking*.

The other type of extreme users is "dummy users." A dummy user would always select top ranked cluster and top ranked document to view. Such a user would just passively follow a system's ranked result. In real web search, it is found that a user's behavior has a view bias and clickthrough can be biased according to the presented ranking order by a search engine [6]. The user tends to view or click on documents from the top. As in the case of a smart user, we can also define *dummy baseline, dummy adaptive*, and *dummy reranking* similarly; the main difference is that the dummy user always chooses the top ranked cluster by the system to view and always chooses the top-ranked document within a cluster to view.

In the real world, a user can be considered as a mixture of smart user and dummy user. That is, on some occasions, the user can intelligently pick the best cluster to view, i.e. apply the strategy of the smart user; on other occasions, the user is influenced by the ranking of search engine and would simply open the top cluster to view, i.e., apply the strategy of the dummy user. We can use the probability of being smart user, which varies from 0.0 (dummy user) to 1.0 (smart user) to control the user interaction behavior.

5.2 Experiment Design

We use TREC8 ad hoc track data set (query topics 401-450) for empirical evaluation. For each query, we use vector space retrieval model to obtain baseline retrieval results. We then use K-Medoids clustering algorithm to cluster the top 100 documents into 6 clusters. In order to evaluate the performance of adaptive clustering representation, we employ a *linearization* method to "convert" any clustering results into a *perceived* ranked list. Specifically, The clusters are first sorted by either the percentage of relevant documents. or the average retrieval score of documents in the cluster, depending on whether a smart user or a dummy user is assumed, respectively. In each cluster, the documents will be sorted by the retrieval scores. The idea is to simulate the perceived order of documents by a user when the user is browsing a clustering result. This way, we can evaluate a clustering result in the same way as evaluating a regular ranked list of result. We use the mean average precision (MAP), precision at 0.1 (pr@0.1) and 0.2 (pr@0.2) recall levels, and precision at top 10 (pr@10d) and 20 (pr@20d) documents as the evaluation metrics.

5.3 Experiment Results

Reranking Row 3-8 of Table 1 shows the retrieval performance of the experiment results for two types of extreme users at different stages, when we use 6 documents in the cluster and 20 terms in the feedback method and the interpolation coefficient is 0.5. We can see that for the smart user, the adaptive clustering strategy is effective and the performance of smart baseline is much better than that of baseline (we assume the user can smartly explore the results according to the cluster labels). For the smart user, smart reranking is apparently better than smart adaptive and smart baseline while smart adaptive has a better pr@10d than smart baseline does. For the dummy user, however, the adaptive clustering strategy appears to be ineffective. The dummy baseline is not as good as the baseline, which means clustering presentation is not effective for dummy users. Dummy adaptive and dummy reranking have similar performances to dummy baseline. Thus the adaptive clustering representation is generally not effective for the dummy user.

Cluster Regrouping In the cluster regrouping strategy, when the smart user selects one cluster, we will merge other clusters into one cluster so that there will be only two clusters – the active cluster and the unopened cluster. Immediately after the smart user selects a cluster to view, the retrieval system will update the user's information need and use the updated query to rerank the documents in each cluster. We call this result

Table 1. Experiment Results of Simulation Study.

Method	MAP	pr@0.1	pr@0.2	pr@10d	pr@20d
baseline	0.230	0.459	0.355	0.398	0.356
smart baseline	0.283	0.529	0.413	0.531	0.465
smart adaptive	0.282	0.559	0.418	0.539	0.470
smart reranking	0.294	0.580	0.428	0.551	0.467
dummy baseline	0.205	0.410	0.324	0.357	0.318
dummy adaptive	0.196	0.414	0.324	0.349	0.326
dummy reranking	0.202	0.418	0.331	0.353	0.320
regroup adaptive	0.261	0.557	0.401	0.533	0.45
regroup reranking	0.270	0.578	0.407	0.539	0.453
promotion adaptive (q)	0.282	0.561	0.421	0.537	0.457
promotion reranking (q)	0.293	0.583	0.434	0.545	0.460
promotion adaptive ($q^{'}$)	0.287	0.575	0.428	0.527	0.471
promotion reranking($q^{'}$)	0.291	0.582	0.428	0.549	0.452

the *regroup adaptive*. When the user selects the best document to view, we can then rerank the documents in each cluster; the corresponding result will be called *regroup reranking*. Row 9-10 of Table 1 shows the retrieval performance of regroup adaptive and regroup reranking. We find that the retrieval performance of regroup adaptive and regroup reranking is not as good as the corresponding smart adaptive and smart reranking.

Near Miss Promotion When we apply the promotion strategy to select a subset of documents from clusters other than the best cluster, we can use the original query q to rank and select documents. We can also use the updated query $q^{'}$, which is interpolated with the best cluster term vector, to rank and select documents. We tried both queries to promote documents. We promote one document from each cluster other than the best cluster and append it to the bottom of the best cluster. Row 11-14 of Table 1 shows the experiment results using this near miss promotion strategy. The first two rows show the experiment results using the original query to promote the documents and the third and fourth row show the experiment results using the updated query to promote the documents. We can see that promotion reranking consistently has better retrieval performance than the promotion adaptive strategy. The better performance of promotion reranking over promotion adaptive clearly comes from reranking documents based on the viewed document; this observation is consistent with what we observed in the performance comparison of smart adaptive and smart reranking.

6 User Study

6.1 Experiment Design

We further conduct a user study by deploying two clustering systems to real users and study whether adaptive clustering strategy can bring better user experience in interacting with the search results than the static clustering strategy. Among the four adaptive clustering strategies, the strategy of promoting "near-miss" strategy has shown some promising results. Promoting "near-miss" strategy dynamically changes cluster structure, so it is likely to make a difference in the interface than simply reranking documents within a cluster. Thus we will use this strategy in the adaptive clustering system. In addition, we will also collect data from real user interactions and questionnaires to study whether there are other factors which may affect adaptive clustering strategies. For example, we can study whether the familiarity with topics has an impact on a user's search experience.

We implement the clustering result presentation functionality in the UCAIR toolbar [14], which is an Internet Explorer (IE) plugin like Google toolbar. The adaptive clustering strategy is also implemented in the UCAIR toolbar. Thus we evaluate two systems with clustering result presentation, i.e. Adaptive System (AS) and Static System (SS). There is a menu of UCAIR toolbar, from which subjects can select to control clustering strategies (AS or SS) they use. The clustering interface is as shown in Figure 1.

Fig. 1. Clustering Interface of Search Results

We randomly select 6 query topics from TREC8 ad-hoc track topics. All of these query topics are informational query. All subjects will just use the same title of each topic as the query to do the web search using one of clustering systems. We randomly divide 6 query topics into 2 groups. Each subject will use System SS to search 3 query topics and System AS to search the remaining 3 query topics. We vary the order of query topics, topic groups, and systems to remove the bias of the order and combination of system and query topics.

After the subjects submit title query to UCAIR toolbar, the UCAIR toolbar will return clustered results to the user by clustering top ranked 100 documents from Google into 6 clusters using K-Medoid clustering algorithm. We use the centroid document to represent each cluster. Subjects browse clusters and click one cluster to view snippets(title, summary, and URL) of documents which belong to the selected cluster. Subjects browsed document snippets and then click the most interesting one to view the content of the document. If it is relevant, subjects will save it on the local disk. Subjects can go back to document list of the selected cluster or go back to cluster interface to select other clusters. There is no restriction about how many clusters the user can view.

For each topic, we ask subjects to find as many relevant documents as possible and learn as much knowledge as possible about each topic in 15-minute period. Subjects are instructed not to save documents hastily.

6.2 Data Collection

We recruit 29 subjects to participate in the user study by posting advertisement at subject recruitment website of the department of psychology. We first conduct a pilot study for 3 users. After the pilot study, we polish the experiment design including questionnaires and procedure, topics, and system implementation. During the formal user study, we ask 24 paid subjects (nearly all of them are undergraduate) to participate the user study. At each session, at most 2 subjects participate in the user study. The user study are conducted in one week. All subjects spend 1.5 to 2 hours on the user study. After the user study, we collect saved relevant documents and questionnaires. For each topic, subjects save some relevant documents on the local disk.

6.3 Experiment Results

Overview We collect user study data (questionnaires, log, and saved documents) from 24 subjects. There are 12 experiment settings, varying the order of the systems, topic groups, and topic orders. All but one subjects are between 16 and 25 years old. There are 16 female and 8 male subjects. Subjects have very diverse majors including nursing, social science, business, physical science, and engineering. All of them use computer daily and have high level expertise with searching.

We collect the exit questionnaires and compare two systems by overall experiences including difference of two systems, helpfulness of systems in completing tasks, easiness of learning to use, easiness of using, and overall preference. The results are listed as Table 2.

From Table 2, we can see that System AS is a little better than SS in the aspect of being helpful in completing tasks and easy to use. Both systems are equally easy to learn since their interface is nearly identical. For the overall preference, System AS seems to be better than System SS. However, there is no clear indication that System AS is better than System SS.

Table 2. Comparison of Overall Experience of Two Clustering systems

Comparison	AS is better	SS is better	No difference
Helpfulness	10	6	8
Easy to Learn	1	3	20
Easy to Use	7	2	15
Overall Preference	11	5	8

Topic Familiarity Factor The number of saved documents may be related with the topic familiarity level of subjects. Thus we compute the average and standard deviation of each topic familiarity level, which varies from 1 (not at all) to 7 (extremely familiar). We also compute the average number of saved document and standard deviation for each topic without differentiating clustering systems used, shown in Table 3.

Table 3. Topic Familiarity and Number of Saved Documents

Topic	1	2	3	4	5	6
Topic Familiarity Avg	3.42	2.71	3.79	1.21	1.21	1.79
Saved Document Avg	8.25	5	6.92	5.33	5.92	5.25
Topic Familiarity Std	1.35	1.27	1.47	0.41	0.72	1.25
Saved Document Std	4.44	3.39	3.61	1.99	2.76	2.69

From Table 3, we can clearly see that topic 1 (Parkinson disease treatment), 2(wildlife preserve poaching impact), and 3(curbing population growth) are more familiar to subjects than topic 4 (legal Pan am 103), 5 (Schengen agreement border control), and 6 (new steel production). However, there is no apparent correlation between the topic familiarity level and number of saved documents.

7 Conclusion and Future Work

In this work, we explore adaptive clustering presentation in interactive information retrieval and propose four adaptation strategies (reranking documents based on a selected cluster, reranking documents based on a viewed document, merging unselected clusters, and promoting “near-miss” documents) to improve clustering results based on a user’s implicit feedback information.

We propose a stochastic way to simulate a user’s browsing behavior and propose a method for evaluating clustering results quantitatively based on user simulation. We evaluate the proposed adaptation algorithms with simulation experiments and show that adaptive clustering, especially reranking of documents based on viewed document is effective for smart users who would intelligently identify and view a high precision cluster and pick a relevant document to view, though such strategies are not effective for dummy users who simply follow a system’s ranking of clusters and documents.

We further conduct a user study to see if an adaptive clustering system can bring improved search utility and/or experience to users, compared with a static clustering system. The results show that there is generally no significant difference between the two systems from a user's perspective. Specifically, more users say that they like the adaptive system better but users saved more relevant documents with the static interface than with the adaptive interface. Overall, our study shows that adaptive clustering has a good potential for improving search utility for users, but a user may not perceive any significant difference in the system.

There are two particularly interesting directions to explore. One direction is to look into other factors to affect the user experience. Although the user study comparing one method of adaptive clustering to static clustering showed no significant user performance or preference differences, this could be due to a variety of factors which we were unable to investigate in this study. Such factors include suitability of the clustering technique to the specific retrieval task; combining user behavior evidence for ranking as well as clustering; and combining several adaptation strategies, rather than using only one. The other one is to do a larger-scale of user study with more topics and multiple iterations of interaction to draw more reliable conclusions.

References

1. C. Carpineto, A. D. Pietra, S. Mizzaro, and G. Romano. Mobile clustering engine. In *ECIR*, pages 155–166, 2006.
2. D. R. Cutting, J. O. Pedersen, D. Karger, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of SIGIR 1992.*, pages 318–329, 1992.
3. S. T. Dumais, E. Cutrell, and H. Chen. Optimizing search by showing results in context. In *Proceedings of CHI 2001*, pages 277–284, 2001.
4. P. Ferragina and A. Gulli. A personalized search engine based on web-snippet hierarchical clustering. In *Proceedings of WWW 2005*, pages 801–810, 2005.
5. M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of SIGIR 1996*, pages 76–84, 1996.
6. T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting click-through data as implicit feedback. In *Proceedings of SIGIR 2005*, 2005.
7. L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons, 1990.
8. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.
9. S. E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33(4):294–304, 1977.
10. S. E. Robertson and S. Walker. Okapi/Keenbow at trec-8. In *Proceedings of TREC of 1999*, 1999.
11. J. J. Rocchio. *Relevance Feedback Information Retrieval.*, pages 313–323. Prentice-Hall, 1971.
12. G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communication of ACM*, 18(11):613–620, 1975.
13. X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proceedings of SIGIR 2005*, pages 43–50, 2005.

14. X. Shen, B. Tan, and C. Zhai. Implicit user modeling for personalized search. In *Proceedings of CIKM 2005*, pages 824–831, 2005.
15. R. W. White, I. Ruthven, J. M. Jose, and C. J. van Rijsbergen. Evaluating implicit feedback models using searcher simulations. *ACM Transaction of Information System*, 23(3):325–361, 2005.
16. O. Zamir and O. Etzioni. Grouper: A dynamic clustering interface to web search results. In *Proceeding of WWW 1999*, 1999.
17. C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR 2001*, 2001.