Relevance Feedback with a Small Number of Relevance Judgements: Incremental Relevance Feedback vs. Document Clustering

Makoto Iwayama
Central Research Laboratory, Hitachi, Ltd.
Hatoyama, Saitama 350-0395, Japan
iwayama@harl.hitachi.co.jp

Abstract

The use of incremental relevance feedback and document clustering were investigated in an relevance feedback environment in which the number of relevance judgements was quite small. Through experiments on the TREC collection, the incremental relevance feedback approach was found not to improve the overall search effectiveness. The clustering approach was found to be promising, although it sometimes over-focuses on a particular topic in a query and ignores the others. To overcome this problem, a query-biased clustering algorithm was developed and shown to be effective.

1 Introduction

In a relevance feedback environment, a system retrieves documents that may be relevant to a user's query. The user judges the relevance of one or more of the retrieved documents and these judgements are fed back to the system to improve the initial search result. This cycle of relevance feedback can be iterated until the user is satisfied with the retrieved documents.

One straightforward assumption we can make here is that the greater the amount of feedback from the user to the system, the better the search effectiveness of the system. Buckley et al. experimentally verified that the recall-precision effectiveness is roughly proportional to the log of the number of known relevant documents [5]. Users consequently are expected to make as many relevance judgements as are needed to obtain satisfactory search effectiveness. However, this expectation is often not met in a highly interactive situation like Internet surfing. In the case of document routing (i.e., document filtering), each query lives a long time and the number of relevance judgements per query can be sufficiently large. This leads to the recent emphasis on massive relevance feedback [4, 14, 11], the main target of which is document routing.

In this paper, we focus on the other extreme case of relevance feedback, in which the number of relevance judgements is quite small, for example, less than 10 or 20 per query. This might be the dominant situation in searching on the Internet because general users are not patient enough to provide hundreds of relevance judgements. In the case of document routing with a large number of available relevance judgements, Allan found that a drop in the number of judgements from thousands to only 10 to 30 decreased effectiveness only about 10% [2]. This indicates that search effectiveness increases greatly when the number of relevance judgements is less than 10 to 30. Our research objective is thus to improve search effectiveness quickly as the number of relevance judgements increases.

For this purpose, we compared two approaches: incremental relevance feedback and document clustering. Both try to help users find many relevant documents having useful feedback information. The incremental relevance feedback approach incrementally reflects the user's relevance judgements, rather than pooling the judgements, and feeding them back all at once. In the ultimate situation, as soon as the user judges one document, the system updates its search criteria based on that judgement. This greedy strategy could boost the number of relevant documents found in the top-ranked portion of a search result. Incremental relevance feedback was originally proposed by Aalbersberg [1] and intensively investigated by Allan for document routing [2]. For a situation with few available relevance judgements, we examined whether this approach can reduce the number of relevance judgements while maintaining high search effectiveness.

The second approach, document clustering, displays retrieved documents in a clustered form rather than in the conventional ranked form. Clustering is applied here aiming to separate the retrieved documents into relevant and non-relevant clusters automatically, expecting the the relevant clusters to be selected satisfactory by users. The clustering approach is widely used in document-browsing interfaces [6, 8]; it helps users to collect relevant documents efficiently [7]. Clustering of retrieved documents also enhances the use of automatic relevance feedback (pseudo feedback) [3]. We systematically evaluated the effect of clustering in our relevance feedback environment, focusing on the relationship between search effectiveness and the number of relevance

SIGIR 2000 7/00 Athens, Greece
© 2000 ACM 1-58113-226-3/00/0007...$5.00

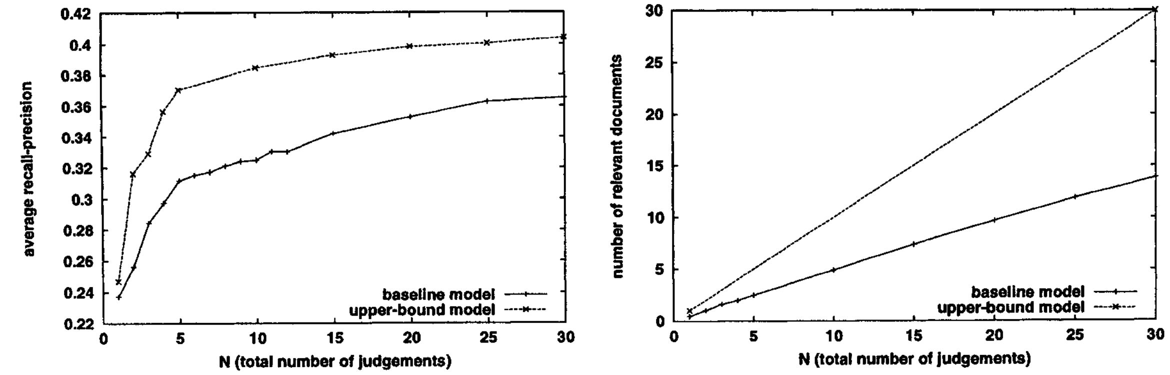

Figure 1: Reference models: search effectiveness (left) and number of relevant documents found (right)

judgements. Because clustering sometimes divided the retrieved documents independently of the user's query, we modified the original clustering algorithm to incorporate the query bias and investigated the effect of this query-biased clustering.

2 Reference Models

We used the TREC collection in our experiments. Documents were taken from TREC Disks 1&2, which contain 742,709 documents. Fifty initial queries were constructed from topics 101 through 150 by extracting only the "title" and "desc(description)" fields for each topic. The relevance judgements in the TREC collection were used to approximate users relevance judgements.

The retrieval model used was the well-known vector space model. We used the Lt.Lnc term weighting formula developed by Singhal et al. [13].

The following modified Rocchio formula was used to update the term weight of a query.

$$Q_t^{new} = \alpha Q_t^{old} + \beta \frac{1}{|\text{rel docs}|} \sum_{\text{rel docs}} wt_t - \gamma \frac{1}{|\text{nonrel docs}|} \sum_{\text{nonrel docs}} wt_t$$

Updated weight Q_t^{new} was calculated from original weight Q_t^{old} and the weights in judged documents. Parameters α, β, and γ were 8, 16, and 4, respectively, which is the most often used combination in this collection. Because the preliminary experiments did not show any significant improvement by using non-relevant judgements in the formula[1] , we used only the relevant ones.

We assumed two reference models, the baseline model and the upper-bound model, to obtain the reference performances. Each of the reference models first retrieves a ranked list of documents for an initial query. The baseline model investigates the top N documents and extracts those marked as relevant. These relevant documents are fed back to the system to construct a new query. The upper-bound model looks through the same ranked list from top to bottom until finding the N relevant documents which are fed back to the system. The reason we call this model an "upper-bound" model is that it is based on the assumption that the list is perfectly ranked, that is, only relevant documents come at the beginning of the list so that users always encounter relevant documents first. In the baseline model, on the other hand, users may encounter non-relevant documents (as in actual situations), the number depending on the quality of the initial ranking. New queries are constructed using the two reference models and used to generate new ranked lists of documents. The average recall-precision was calculated against these new rankings [2] .

Figure 1 (left) plots the overall average recall-precision for all 50 queries at various points of N. Because the focus here is on a small number of relevance judgements, we only investigated the cases where N was less than 30. Figure 1 (right) plots the number of relevant documents found in the two listings. Using these results, we attempted to approach the upper-bound performance starting from the baseline performance.

3 Incremental Relevance Feedback Approach

3.1 Method

In the original incremental relevance feedback environment proposed by Aalbersberg [1], a system provides a user the single supposedly most relevant document to the present query, and once the user judges the relevance on the document, the system uses the judgement to update the query and recalculates the relevance scores of all the documents based on the updated query. Consequently, the retrieved documents are always ranked

[1] Although non-relevant judgements improved the search effectiveness in almost all the runs, the amount of improvement was small enough compared to that due to the relevant judgements to be ignored. Note that our experiments were not for massive relevance feedback, in which non-relevant judgements contribute much more to improving search effectiveness.

[2] This evaluation method is retrospective, that is the performance is evaluated on the same set of documents which is used for relevance judgements. Because this is the real situation of interactive relevance feedback, we simply used the closed set of documents and did not use another set of documents for evaluation

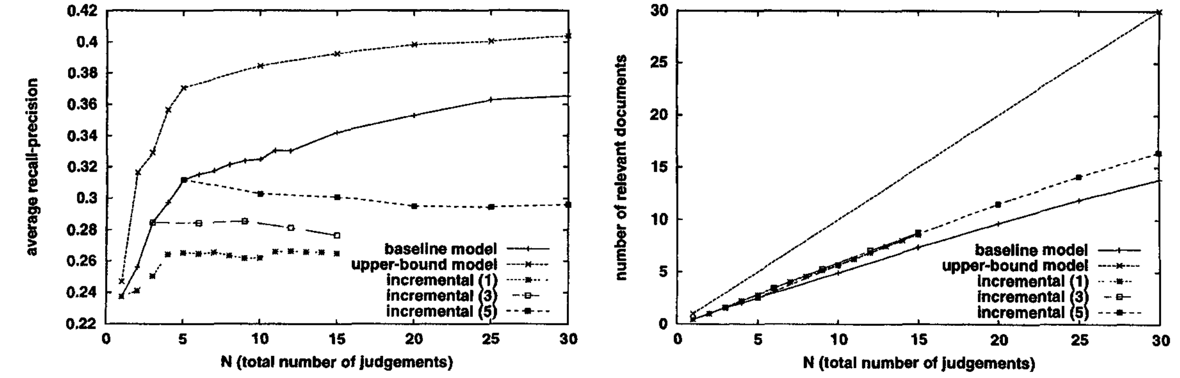

Figure 2: Incremental relevance feedback approach: search effectiveness (left) and number of relevant documents found (right)

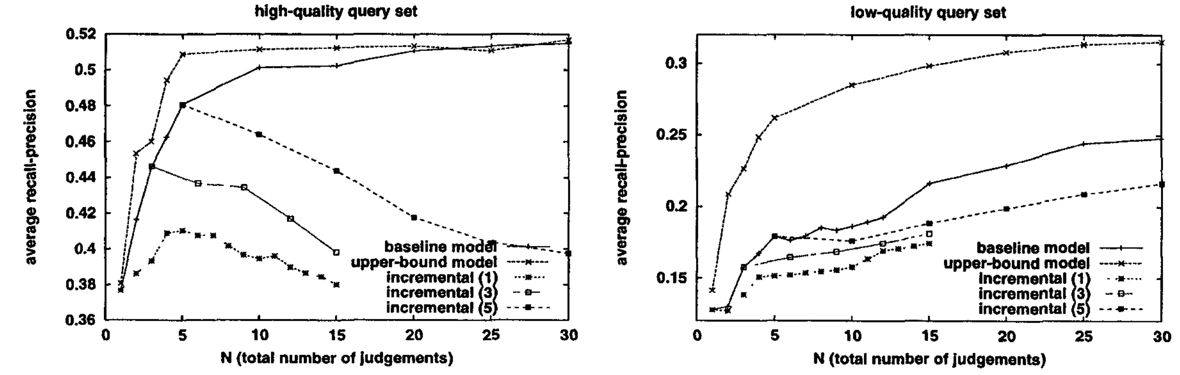

Figure 3: Incremental relevance feedback approach: search effectiveness for high/low-quality query set

based on the user's judgements so far. This greedy strategy can find more relevant documents compared to the case without incremental updating.

Consider the situation where the user finds a relevant document in a ranked list. There are two possibilities for the next step: one is to keep the original ranking and find the next relevant document (i.e., the baseline model) and the other is to update the original ranking by feeding back the relevant document just found and find the next one from the updated ranking (i.e., the incremental relevance feedback approach). Because the system improves the search effectiveness as the number of provided judgements grows, as we can see in Figure 1, the user will likely find more relevant documents in the updated ranking than in the original ranking.

Aalbersberg did not experimentally verify this assumption [1] [3] . Allan intensively analyzed the incremental relevance feedback approach [2], but the task was batched document routing, in which judgements are incrementally provided to the system from a static set of judgements. We are interested in the original situation proposed by Aalbersberg: the source of judgements is incrementally updated based on the latest query.

3.2 Results and Discussion

In Figure 2 (left), the results for the incremental relevance feedback approach are shown along with those for the reference models. The N is the cumulative number of judgements incrementally provided to the system. In addition to the original incremental feedback approach, we also ran it with buffering, in which judgements were pooled temporally and fed back to the system all at once. The buffering was done at factors of 3 and 5.

Unfortunately, the results were disappointing. The average recall-precision of all these runs was greatly inferior to that of the baseline model. In the run with buffer size 1, only the first few feedback cycles were effective. In the runs with buffer sizes of 3 and 5, all feedback cycles did not improve the rankings for each. This is not because of a small number of relevant documents found. Figure 2 (right) shows that the incremental relevance feedback approach found more relevant documents than the baseline model, as expected.

[3] His experiments were conducted using single-feedback iteration

It sounds strange that the incremental relevance feedback approach cannot outperform the baseline model despite finding more relevant documents. One possible reason is that, in the incremental feedback approach, most of the relevant documents newly found in each feedback cycle are duplicates of previously found relevant documents or are related to a sub-topic of the present query. This is because the judgements in each feedback cycle are made for only the top-ranked documents. In the baseline model, the judged documents came from a static set linked to the initial query only, and they may be related to a greater variety of topics as N grows, including more marginal ones. We can glimpse this phenomenon in Figure 2 (left): the buffered runs with the buffer sizes of 3 and 5 did not improve their initial rankings from the standpoint of the overall search effectiveness.

To clarify this effect, we divided the 50 initial queries into two sets and investigated the search effectiveness for each. The first query set, a "high-quality" query set contained queries that retrieved many relevant documents in the top ranking (more than 15 relevant documents in the top 30 documents). These queries should contain most of the topics in the information need. The other set, a "low-quality" query set, contained queries having poorer retrieval power (15 or fewer relevant documents in the top 30 documents). Topic id's in each query set are listed as follows.

high-quality set	low-quality set
106, 107, 108, 109, 110, 111, 112, 115, 118, 123, 130, 132, 133, 134, 135, 136, 137, 142, 145, 146, 148, 150	101, 102, 103, 104, 105, 113, 114, 116, 117, 119, 120, 121, 122, 124, 125, 126, 127, 128, 129, 131, 138, 139, 140, 141, 143, 144, 147, 149

The average precision for the top 30 documents was 0.7500 for the high-quality query set and 0.2345 for the low-quality one.

Figure 3 shows that the incremental updating for the low-quality set improved the ranking incrementally, but the search effectiveness was still less than that of the baseline model. On the other case, for the high-quality set, the performance at every feedback cycle, except for the first few ones for buffer size 1, steeply fell after the initial search, although the number of provided judgements increased.

Based on these results, the incremental relevance feedback approach is not appropriate for increasing overall search effectiveness; however, it might be useful for finding similar documents on a more specific topic because of its greedy hill-climbing algorithm. We also found that simply feeding a large number of relevant documents back to a system is not enough for the system to cover all the topics for the required information. The overall performance depends on the variety of relevant documents provided, and this research issue is closely related to the sampling issue discussed by Lewis and Gale [10].

4 Clustering Approach

4.1 Method

The clustering approach is based on the "cluster hypothesis" [16], which assumes that the documents relevant to an information need are similar to each other. If this assumption is correct, clustering can be used to separate the set of retrieved documents into relevant and non-relevant clusters. Through successive selection of relevant clusters, users can find many relevant documents at little cost because they do not have to consider the non-relevant documents located in the unselected clusters.

Following this line, many researchers have applied clustering to the set of retrieved documents. The "Scatter/Gather" method focuses on relevant topics efficiently by iteratively applying clustering/selection to the retrieved documents [6]. Clustering of the initial search results was reported to be effective for subsequent pseudo relevance feedback [3]. Evans et al. showed the usefulness of clustering in a relevance feedback environment through a user study. They found that the clustered representation of retrieved documents to users resulted in improved search effectiveness [7]. Following these results, especially those of Evans et al., we systematically investigated the relationship between search effectiveness and the number of relevance judgements fed back to the system. It is useful to investigate this relationship because, as we saw in the previous section, a larger number of relevant documents does not always improve the overall search effectiveness.

In the experiments, we first retrieved the top 150 documents from each of the 50 initial queries. These 150 documents were grouped into 5 clusters. While any clustering algorithm can be used, we used the probabilistic algorithm proposed by Iwayama and Tokunaga [9]. To select the best clusters, we used the "DENSITY" strategy [12], in which the clusters are sorted by the proportion of relevant documents they contain. Within each cluster, the documents are sorted by their relevance score for the initial query. From the top ranked cluster, we extracted the top N documents and those marked relevant were fed back to the system. If N exceeded the number of documents in the top ranked cluster, we moved to the second ranked cluster, and so on.

There remains room for discussion of the cluster selection method. Using the "DENSITY" strategy assumes that users can select clusters containing many relevant documents. This would be a rather strong assumption, or it would cost much effort for users to select such clusters. Although there is preliminary evidence that "users can select the cluster with the largest number of relevant documents in most cases" [8], this issue requires more investigation.

4.2 Results and Discussion

Figure 4 (left) shows the search effectiveness for all 50 queries. Unlike the incremental relevance feedback approach, the clustering approach gradually improved the search effectiveness as N grew, and it always outperformed the baseline model. Figure 4 (right) shows that the selected best clusters contained more relevant documents than the original ranked list (the baseline model).

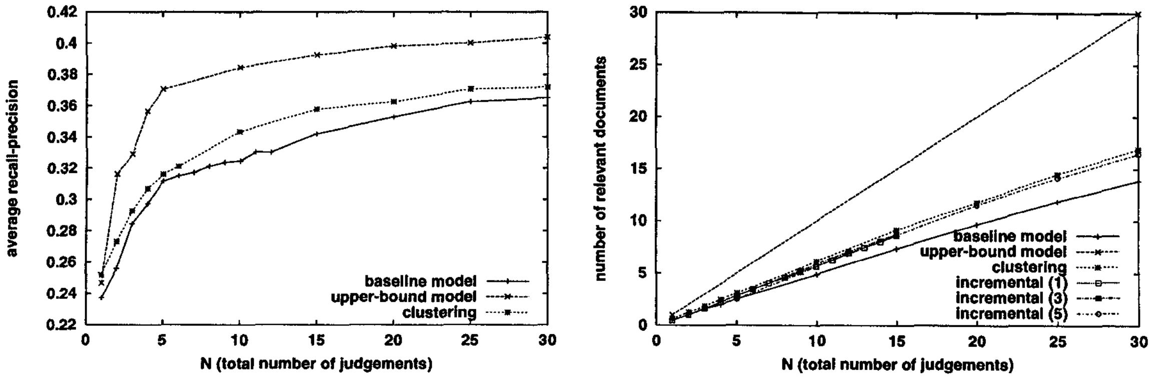

Figure 4: Clustering approach: search effectiveness (left) and number of relevant documents found(right)

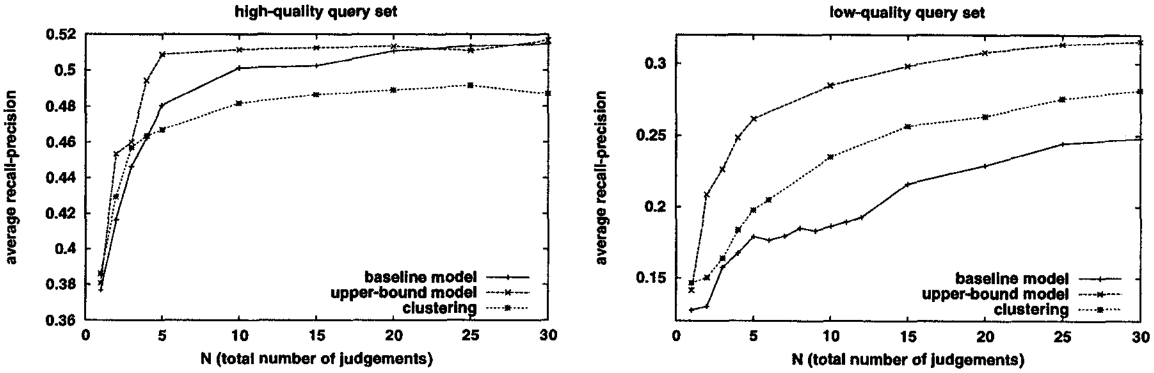

Figure 5: Clustering approach: search effectiveness for high/low-quality query set

This fact confirms the effectiveness of the "cluster hypotheses" in the clustering approach, the reason for the superiority of the clustering approach over the baseline model.

The clustering approach significantly outperformed the incremental relevance feedback approach, although both approaches found almost the same number of relevant documents for the same number of judgements (as seen in Figure 4 (right)). This implies that the two approaches find very different kinds of relevant documents.

As in the case of the incremental relevance feedback approach, we investigated the search effectiveness for two query sets: the high-quality one and the low-quality one. As shown in Figure 5, for the high-quality set, the clustering approach did not outperform the baseline model for most values of N. This is because clustering divides retrieved documents, most of which are relevant in this case, into several clusters and selects only one or two of these clusters. It thus focuses too much on a particular topic of relevance. This reduces the overall search effectiveness. For the low-quality query set, on the other hand, the set of retrieved documents has a relatively small number of relevant documents, so clustering and successive cluster selection finds the best and only topic of relevance, which makes the clustering approach superior to the baseline model, as shown in Figure 5. Although there are several remedies for the over-focusing problem, such as skimming off the best documents from the best clusters, we consider another approach in the next section.

5 Query-biased Clustering

The experimental results for the high-quality query set revealed the importance of the query, especially those queries that retrieve many relevant documents. The clustering approach is no more effective in this case than the baseline model, which primarily uses the document set retrieved from the initial query. In this section, we modify the clustering algorithm to consider this query effect and try to improve its performance, especially for the high-quality query set.

In the original clustering approach, the query determines the scope of documents for clustering, but the clustering algorithm itself is independent of the query. We incorporate the effect of the query directly into the clustering algorithm in the following way. Our cluster-

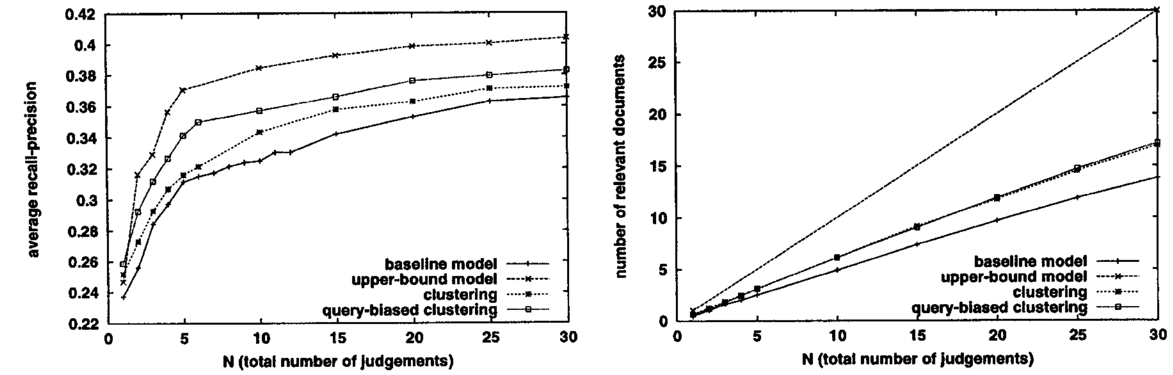

Figure 6: Query-biased clustering approach: search effectiveness (left) and number of relevant documents found (right)

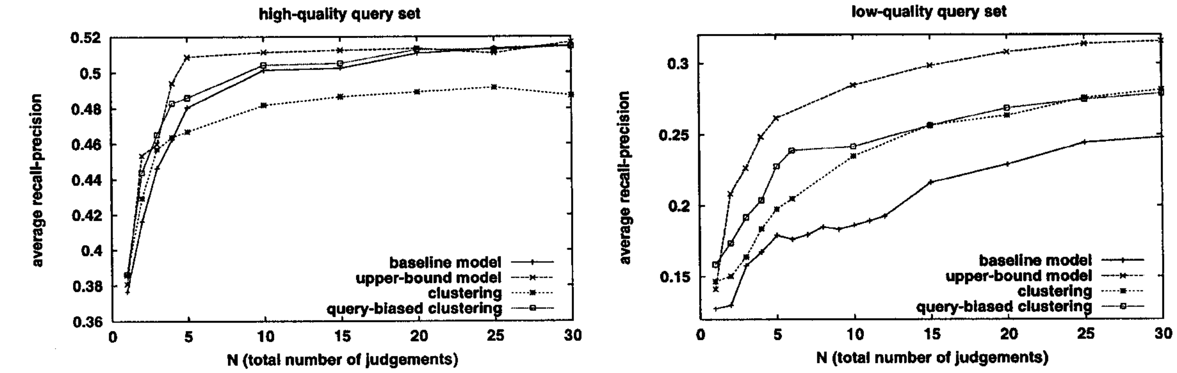

Figure 7: Query-biased clustering approach: search effectiveness for high/low-quality query set

ing algorithm calculates $P(C|d)$, the probability of cluster C given the information of document d, and agglomeratively constructs the set of clusters that maximizes $\prod_C \prod_{d \in C} P(C|d)$. We add the information of query q to the conditional part of $P(C|d)$ and obtain $P(C|d,q)$. More precisely, the conditional part of $P(C|d,q)$ is calculated by adding the term weights for q to the corresponding term weights for d. This simple method of raising the importance of terms occurring in a query can be applied to any conventional clustering algorithm, making it a query-biased one.

In the experiments, we only replaced the original clustering algorithm with the query-biased one. Figure 6 (left) shows that the query-biased clustering improves the original clustering approach significantly. The amount of improvement against the baseline model is also significant. These improvements are mainly due to large improvements for the high-quality query set (see Figure 7), where the use of query-biased clustering had a comparative advantage over the baseline model, which uses the initial query at most. Even with only a few judgements, the query-biased one was better than the baseline. For the low-quality set, the query-biased approach outperformed the original clustering approach, especially for small N. This confirms that the "clustering hypotheses" works effectively in query-biased clustering. Figure 6 (right) shows that both clustering approaches found about the same number of relevant documents with the same number of judgements.

In summary, the use of query-biased clustering avoids the over-focusing problem seen in the original clustering approach while keeping the advantage of the "cluster hypothesis."

6 Conclusion

We have discussed the use of incremental relevance feedback and document clustering in a relevance feedback environment in which the number of relevance judgements was quite small. Through experiments on the TREC collection, we showed that the incremental relevance feedback approach does not improve the overall search effectiveness. The clustering approach was shown to be promising, although it sometimes over-focuses on a particular topic in a query and ignores the others. To overcome this problem, we introduced a query-biased clustering algorithm and demonstrated its usefulness.

While the results for the incremental relevance feedback approach were disappointing, this approach might be useful for quickly focusing on a particular aspect of the information need. We did not verify this advantage quantitatively, but only observed its effect in the experiments. More intensive investigation of this issue is necessary.

Studies are needed to determine whether actual users can find the best clusters easily in the clustering approach. Although a preliminary study by Hearst and Pedersen confirmed this assumption [8], there is no experimental support for query-biased clustering. Through personal observation in the experiments, we believe that query-biased clustering can separate the relevant from the non-relevant documents more clearly than the original clustering, and this should make cluster selection by actual users easier.

Lastly, query-biased clustering can be applied to other search environments, Scatter/Gather-style search interface [6], query-biased summarization [15], etc. Query-biased clustering would be effective because it can produce different sets of clusters according to the bias given by the user. This enables a user to obtain a tailored set of clusters that reflects his/her information needs more precisely.

Acknowledgment

This research is supported in part by the Advanced Software Technology Projects under the auspices of IPA (Information-technology Promotion Agency), Japan.

References

[1] I. J Aalbersberg. Incremental relevance feedback. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 11–22, 1992.

[2] J. Allan. Incremental relevance feedback for information filtering. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 270–278, 1996.

[3] C. Buckley, M. Mitra, J. Walz, and C. Cardie. Using clustering and SuperConcepts within SMART: TREC 6. In *Proceedings of the Sixth Text REtrieval Conference (TREC-6)*, 1998.

[4] C. Buckley and G. Salton. Optimization of relevance feedback weights. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 351–357, 1995.

[5] C. Buckley, G. Salton, and J. Allan. The effect of adding relevance information in a relevance feedback environment. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 292–300, 1994.

[6] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/Gather: A cluster-based approach to browsing large document collections. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–329, 1992.

[7] D. A. Evans, A. Huettner, Tong X., P. Jansen, and J. Bennett. Effectiveness of clustering in ad-hoc retrieval. In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, 1999.

[8] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996.

[9] M. Iwayama and T. Tokunaga. Cluster-based text categorization: A comparison of category search strategies. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 273–280, 1995.

[10] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

[11] R. E. Schapire, Y. Singer, and A. Singhal. Boosting and rocchio applied to text filtering. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 215–223, 1998.

[12] H. Schütze and C. Silverstein. Projections for efficient document clustering. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1997.

[13] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, 1996.

[14] A. Singhal, M. Mitra, and C. Buckley. Learning routing queries in a query zone. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 25–32, 1997.

[15] A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2–10, 1998.

[16] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.