

IMPROVING RELEVANCE FEEDBACK WITH UNBIASED ESTIMATE OF USER'S INFORMATION NEED

Yunjie Xu
School of Management
Syracuse University
yuxu@sry.edu

Abstract

Relevance feedback is an effective and widely accepted method in information retrieval to improve performance. Relevance feedback generally uses an adaptive learning method to estimate the user's information need. In this research, we propose an alternative two-stage sampling method to obtain an unbiased estimate of the user's information need. Our estimate shows not only improved retrieval performance, but also better prevention of query drift, which troubles traditional relevance feedback. We also give theoretical justification and empirical support for this method.

Keywords: Two-stage sampling method, information retrieval, relevance feedback, automatic relevance feedback, user's information need.

INTRODUCTION

It is already a cliché to say that the Internet age is an age of information overload. Since large amounts of information exist in the form of text documents, effective information retrieval technology becomes an immediate need for both information producers and consumers. Some applications of information retrieval technique in business environment can be: (1) use of the Internet for business intelligence, (2) dynamic generation of a web site that accommodates the information need of the current user, and (3) text retrieval techniques to manage the corporate knowledge base and know-how that are generally text files. All of these applications demand high performance information retrieval capacity.

If we view an information retrieval system as consisting of a searcher, a search mechanism, and information, information retrieval effectiveness can be improved through any of them. Better information organization, such as XML (Egnor and Lord 2000) and ontology (Fensel et al. 1998), can impose more structure on documents and thus improve retrieval accuracy. Better understanding of the user's searching behavior (Lieberman 1995; Robins 2000) can also help us design better personal information agents. Nevertheless, most research in information retrieval tries to improve the search mechanism, where the objective is to find a better set of relevant documents given the information need. The goal of this study is to improve the search mechanism with a new relevance feedback method.

An information retrieval system usually represents the document as a vector of terms (keywords or phrases). Let \mathbf{V} be the vocabulary of all terms and t being a term in the vocabulary, then a document is such a vector:

$$d_i = (t_{i1}, t_{i2}, \dots, t_{im}), t_{ji} \in \mathbf{V}$$

Each term is an element of the vector and is assigned certain weight to represent its importance. For example, t_{ji} can be frequency of a term in the document.¹ Queries are quantified in the same way. Therefore, to measure how relevant a document is to a query, a vector similarity measure, such as cosine, can be computed (Salton and McGill 1983). Since such a mechanism depends on how accurate a user describes his/her information need, it is deemed beneficial to revise the user's initial query. Relevance feedback is such a mechanism of revision. When an information retrieval system returns a list of documents ranked by similarity to a query, relevance feedback method asks the user to read some of them and classify them as relevant or not. With the classified documents, the system revises the user's original query and carries out a second round run. Relevance feedback is generally accepted with reported success (Salton and Buckley 1995). The purpose of relevance feedback is to revise the user's query so that it better represents the user's information need. The query revision technique used is adaptive learning.

However, it is still not clear that such an adaptive learning method will make the best use of the classified documents and generate an unbiased estimate of the user's information need. There are other problems like parameter setting (Salton and Buckley 1995) and query drifting (Buckley et al. 1994) that plague relevance feedback. Rather than using adaptive learning, we propose a two-stage sampling method to generate an unbiased estimate of the user's information need. Our method can better estimate the user's information need and better prevent the revised query from drifting away from the user's original intention. This study contributes to the information retrieval community an alternative and more effective query revision method. The paper is organized as follows. The next section gives the literature review. The proposed method and its theoretical property are discussed in the third section. Empirical support is then given and the conclusions are presented.

LITERATURE REVIEW

Research Question

Relevance feedback improves capture of the user's information need through query revision. It is first necessary to define the user's information need. Rather than getting into the behavioral and psychological side of the user's search behavior, this study assumes that the user has a fixed information need and is able to classify any document (as relevant or not) correctly. We also assume that all of the relevant documents in the document database are enough to satisfy the user's need. If documents are represented by quantified vectors of terms, we define the user's information need as the average vector of all relevant documents in the database.

Given the user's information need, the search mechanism is to find documents that best meet the need. Unfortunately, it is generally believed that the user's query is, at best, a rough representation of the user's true information need. Therefore, the search mechanism needs to address two issues: (1) how to better capture the user's information need and (2) how to find a set of documents that best meet the need. Relevance feedback, along with a whole stream of research in query expansion, is aimed at the first issue (Efthimiadis 1996). This study has the same purpose.

Relevance Feedback and Query Expansion

The average length of a search query on the Internet is two words. To better capture the user's information need, two types of approach are usually adopted. First, additional key terms can be added to the original query. Second, weights for terms in the query can be revised to reflect their importance. In this section, we review some techniques to expand and re-weight the query.

To expand the query with additional terms, synonyms, hypernyms (the super-concept), or words in the same thesaurus can be added, or any match to these terms can be regarded as a match to the original term (refer to Efthimiadis for a detailed treatment). Such expansion is usually done without term re-weighting and demands natural language processing or special knowledge base support. On the contrary, relevance feedback is a method that both expands and re-weights terms.

¹There are several schemes for weighting keywords or terms. A popular one is called *TFIDF* (Salton and McGill 1983). The *TFIDF* acronym comprises two terms: *TF* is the frequency of a term in a text and *IDF* is the inverse of document frequency of a term, which is the total number of documents containing a specific word divided by the total number of documents. Thus, while *TF* measures the representation power of a term in a document, *DF* measures the general discriminatory power of a term in the corpus.

When the retrieval system returns a preliminary set of documents, relevance feedback asks the user to classify the retrieved result and use this information to revise the original query. An adaptive term weighting method is usually adopted to revise query (Rocchio 1965). This method can be described by the equation:

$$q' = \alpha q_0 + \beta \sum_{d_i \in R} d_i - \gamma \sum_{d_i \in I} d_i$$

where q' and q_0 are the revised and the original user query represented as document vectors, and α , β , and γ are coefficients. R and I are relevant and irrelevant document sets identified by the user in the relevance feedback process. Such a revision process can go several iterations.

In this equation, the terms in all relevant documents known so far are used to increase the weight of the corresponding term in the original query, and vice versa. Therefore, the revised query generally gives higher weight to those terms in relevant documents and lower weights to those in non-relevant documents. Since the revised query better represents what the user is looking for in a document, we can use it to recalculate, by some similarity measure, how closely documents match it and, therefore, how well they fit the user's need.

If relevance feedback requires the user to decide which document is relevant, it is usually called real relevance feedback. If a few documents returned by the retrieval system are treated as relevant or irrelevant automatically without user intervention, it is called automatic (pseudo-) relevance feedback (Buckley et al. 1994; Efthimiadis and Biron 1995). In automatic relevance feedback, if only the top few documents are used, it is positive feedback. Since automatic relevance feedback blindly assumes the top few documents are relevant, it might mistake some irrelevant documents for being relevant.

Problems with Relevance Feedback

There are two potential problems that can cause relevance feedback to fail. First, Rocchio's relevance feedback method implies that the user's information need is changing. Therefore, the old weight of a term fades as more feedback arrives. In an iterative and interactive retrieval process, this is a reasonable assumption because the user intention may change over time. While in a single query-retrieval cycle, when the user submits a query only once, it is questionable that we should assume in this way. This problem is particularly obvious with automatic relevance feedback. Because the blindly assumed relevant documents actually can be irrelevant, using them to revise the query causes the query to drift away from the user's original intention. This is the query drift problem.

In the setup of automatic relevance feedback, to prevent query drift, Mitra et al. (1998) and Hearst (1996) proposed some methods to purify the top list by imposing extra filtering criteria, so that the top few are all relevant. For example, documents should have all the keywords to be treated as relevant; the proximity of keywords is important, etc. Allan (1996) proposed an incremental relevance feedback method. These studies to some degree prevent query drift and show some effective ways to get a reasonably good sample of relevant documents without extra user effort. But they did not change the fundamental adaptive learning of relevance feedback.

If the user's information need does not change, theoretically, it is hard to claim that the relevance feedback will give the best estimate of the user's information need. For example, if the average vector of all relevant documents gives the best approximation of the user's information need, the relevance feedback equation will not converge to the average in several runs, because it is almost impossible to set the best values for α , β , and γ , when the size of relevant documents is unknown beforehand. Based on this observation, the second problem with relevance feedback is the difficulty in setting the coefficients. Early study suggests that positive feedback from relevant documents should be preferred. Thus β should be greater than γ ; and in automatic positive relevance feedback, $\gamma = 0$. A feedback formula that is better than Rocchio's has been proposed (Buckley et al. 1994; Salton and Buckley 1995). It used $\beta \times 1/\|R\|$ and $\gamma \times 1/\|I\|$ as coefficients in the place of β and γ , where $\|R\|$ and $\|I\|$ are the number of classified relevant and irrelevant documents respectively. This method does not point out a way to set coefficients, but it does try to mitigate the bias introduced by the number of document in relevant and irrelevant samples.

It is our purpose to improve the estimate of the user's information need and prevent query drift, yet introduce minimum extra user effort or system resource. In the following section, we give a detailed description of our method and its theoretical justification.

TWO-STAGE SAMPLING METHOD

The Rationale

When we have a sample—relevant document (classified by user or automatically classified by system)—a statistical estimate could be obtained from the sample as a representation of the user’s information need. We can use the average vector of the sample as our estimate. If the sample of relevant documents is random, it is easy to see that the estimate is better than the original query alone. The problem with such an estimate is that the classified relevant document set is generally not a random sample. The set is a few documents with the highest cosine similarity (or other similarity measure) for a given query. As a result, the estimate obtained is biased. To obtain an unbiased estimate, we first need to introduce another assumption. It is commonly assumed in information retrieval literature that the use of different words is independent. This assumption is more for mathematical convenience than a fact. Yet Salton and McGill (1983) argue that the impact of such simplification is not large. Based on this assumption, if a document is represented by a group of words, the words are independent. The implication of this assumption is that although such a sample of classified relevant documents is not a random sample for terms that appear in the query, it can still be treated as a random sample for other terms that are not in the original query but are going to be included in the revised query. Thus we can construct an unbiased estimate for those terms. In turn, if those additional terms are used as a search query, and a second set of relevant documents is obtained, we can then use that set to obtain an unbiased estimate for terms in the original query.

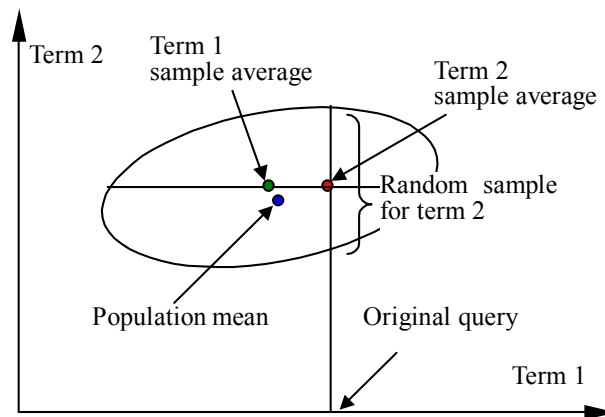


Figure 1. Two-Stage Sampling Method

Schematically, suppose relevant documents have only two terms, term 1 and term 2, following a distribution as depicted in Figure 1. The original query has only term 1. When a sample of relevant documents is returned, we can obtain a sample average for term 2. If we use the sample average of term 2 as a query; we can obtain a second sample of relevant documents, from which we obtain a sample average for term 1. The final estimate of term 1 and term 2 is unbiased and will be closer to the true mean of the population of relevant documents.

The Procedure and Theoretical Justification

The procedure can be formalized as follows:

1. Use the original query to retrieve a sample of relevant documents (user classification may be needed if it is not automatic relevance feedback).
2. Use the sample average of the retrieved relevant documents as an estimate for terms not in original query.
3. Use only these terms as an additional query to retrieve a second sample of relevant documents (user classification may be needed if it is not automatic relevant feedback).
4. Obtain the sample average for terms in the original query.

5. Combine the estimate for terms in the original query and those that are not. Use the combined vector as an estimate of the user's information need to carry out a final run.

We will formally show that our method gives an unbiased estimate and is more accurate than the original query alone.

We represent the document as a vector of words in a fixed vocabulary with TFIDF weighting. Suppose that all of the relevant documents in the document database can satisfy the user's information need. These relevant documents make up our population of relevant documents. Let its mean be μ , variance-covariance matrix be Σ . An estimate of μ , q , is unbiased if the expected value of q , $E(q) = \mu$.

Let q_0 be a vector representing the original query. Let K_1 be the set of relevant documents returned for q_0 (step 1), and let K_2 be the set of relevant documents returned on step 3. Let \bar{K}_1 be the average vector of K_1 excluding terms in q_0 , and \bar{K}_2 be the average vector of K_2 excluding terms in \bar{K}_1 .

Then the final query (estimate of the user's information need) is:

$$q_2 = (\bar{K}_1, \bar{K}_2)$$

Because for each term t in q_2 , we have a random sample, the expected value of term sample average is its mean, $E(\bar{t}_i) = \mu_{ti}$, where $\bar{t}_i \in \bar{K}_1$ or \bar{K}_2 , $i=1 \dots \max(\|\bar{K}_1\|, \|\bar{K}_2\|)$, $\|\bar{K}_i\|$ is the number of elements in the vector, and μ_{ti} is the population mean for the term. Also notice that the deviation for term sample average is $s(\bar{t}_i) = \sigma_i^2 / \|\bar{K}_i\|$, $i=1,2$, where $s(\bar{t}_i)$ is the variance of the sample average for term t and σ_i^2 is the true variance. Based on the above observations, the expectation and covariance matrix of q_2 and $S(q_2)$ are:

$$E(q_2) = \mu$$

$$S(q_2) = \begin{pmatrix} S(\bar{K}_1) & \mathbf{0} \\ \mathbf{0} & S(\bar{K}_2) \end{pmatrix} = \begin{pmatrix} \frac{\sigma_1^2}{\|\bar{K}_1\|} & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \frac{\sigma_m^2}{\|\bar{K}_2\|} \end{pmatrix}$$

Note that because terms are independent, the covariance among terms is zero.

We thus show q_2 is an unbiased estimate of μ . If we further assume that the original query q_0 is a random relevant document, then $E(q_0) = \mu$, and $S(q_0) = \Sigma$. Compared with q_0 , not only is q_2 unbiased, it also has a smaller covariance matrix, which means the estimate is more accurate.

The above method could involve having the user twice supplying samples of relevant documents. This may be a drawback compared to regular relevance feedback that involves the user only once in a single cycle. To avoid extra user effort, the sample of relevant document can be obtained using the top few returned documents in automatic relevance feedback. Because there could be irrelevant documents in the top, we can purify them (see Mitra et al. 1998). In the following empirical test, we use automatic relevance feedback without purification.

EMPIRICAL SUPPORT

Description of Data Sets

We test our procedure on four data sets that have been used in other information retrieval studies. These data sets are available on the Web. A short summary of these data sets is given below.

| Datasets | Description | Sample size | No. of query tested |
|------------------|---|-------------|---------------------|
| Time | <i>Time Magazine</i> full text articles in 1963 | 425 | 60 |
| Medline | Medical literature abstracts | 1033 | 30 |
| Cranfield | Mechanical literature abstracts | 1400 | 225 |
| CISI | Computer Information Science Index abstracts | 1460 | 112 |

Sample size is the number of documents in each data set. For each query, the data set also provides a corresponding list of relevant documents. We thus have a “hard” standard to judge the performance of our search mechanism. Queries are in natural language form.

For each document, we do the regular preprocessing, including stop words, deletion, and Porter’s stemming. When converting a document to a vector, we keep all of the words in it, because the documents are usually short. TFIDF weighting scheme is used with normalization. There are a few questions that we want answered by our experiment:

- How well do the two-stage sampling method and its benchmarks perform?
- How well does the two-stage sampling method capture the user’s information need?
- How well does the two-stage sampling method prevent query drift?

Retrieval Performance

Our first goal is to test the retrieval performance of the proposed method. To achieve the first goal, three benchmarks are tested. The vector space model benchmark uses a standard vector space model with cosine similarity measure. The top 10 relevance feedback benchmark uses automatic relevance feedback with only positive feedback from the top 10 articles. The cut-off feedback benchmark uses automatic relevance feedback with dynamically selected top K documents. The cut-off point is set to 50% of the highest cosine similarity. Any document that scores above that cut-off point is treated as relevant. We suspect that a dynamically selected set of top K documents may perform better than the fixed top 10 relevance feedback and prevent query drift, if the cut-off point is appropriate. The same set of dynamically selected top K documents will be used for our two-stage sampling method.

Here we report the precision-recall measurements and average precision of the two-stage sampling method and benchmarks. The average precision is the average precision over all recall levels, which serves as a rough measurement of the overall performance of an algorithm.

Table 1. Average Precision*

| Datasets | Sample size | No. Of query | Std. Vector space | Top 10 RF | Cut-off RF | Two-stage sampling |
|------------------|-------------|--------------|-------------------|-----------|------------|--------------------|
| Time | 425 | 60 | 77% | 74% | 79% | 84% |
| Medline | 1033 | 30 | 56% | 62% | 62% | 66% |
| Cranfield | 1400 | 225 | 33% | 34% | 36% | 38% |
| CISI | 1460 | 112 | 26% | 26% | 26% | 28% |

*Number in bold face denotes the best performance.

This result shows the performance is decreasing in the order of two-stage sampling method, cut-off relevance feedback, top 10 relevance feedback, and vector space model. To see that the two-stage sampling method is superior to the adaptive learning of relevance feedback, we should notice that, even when they use the same top K documents to estimate the user’s information need, the performance is still significantly different. However, the cut-off relevance feedback has only marginal advantage over top 10 relevance feedback in this experiment.

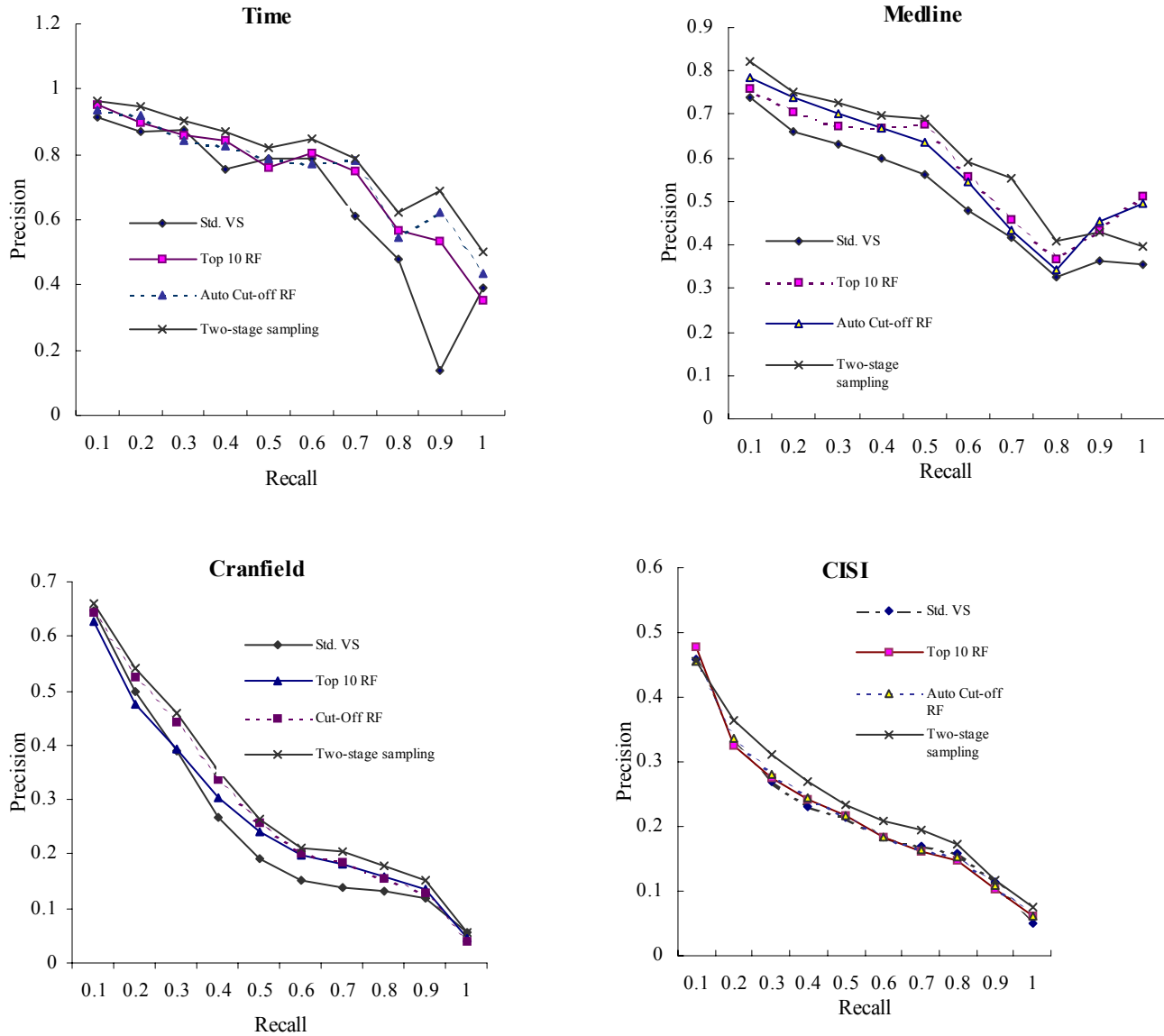


Figure 2. Precision and Recall for Four Data Sets

Estimate Quality

Our second goal is to test estimate quality, i.e., how good our estimate of the user’s information need approximate the user’s true information need, defined as the average of all relevant documents.

To test how an estimate approximates the user’s information need, for each query we measure its cosine similarity with the user’s information need. In the Table 2, we use the average cosine over all the queries as an overall measure of estimate quality.

Table 2 shows the following statistics. The “Avg. $\cos(q_2, \mu)$ ” column is the average of cosine similarity between the two-stage estimate and the user’s true information need. The “Avg. $\cos(\text{RF_CF}, \mu)$ ” column is the average of cosine similarity between the cut-off relevance feedback estimate and the true information need. The value in parentheses is the standard deviation of the corresponding cosine similarity over all of the queries. The “p-value” column is a paired t-test result on the null hypothesis that the two-stage estimate is equal to the estimate from cut-off relevance feedback. It is clear that the average cosines of the two-stage

estimate are significantly better estimates of the true information need than those of the cut-off relevance feedback. Since cut-off relevance feedback performs generally better than top 10 relevance feedback and original query alone, we do not supply additional test data here.

Table 2. Estimate Quality Comparison

| Datasets | Avg. $\cos(q_2, \mu)$ (std. dev.) | Avg. $\cos(\text{RF_CF}, \mu)$ (std. dev.) | p-value |
|------------------|--------------------------------------|--|----------|
| <i>Time</i> | 0.576 (0.238) | 0.4748 (0.318) | 0.029 |
| <i>Medline</i> | 0.5422 (0.1945) | 0.4181 (0.1961) | 1.19E-13 |
| <i>Cranfield</i> | 0.5974 (0.1464) | 0.3353 (0.1273) | 2.75E-22 |
| <i>CISI</i> | 0.3544 (0.1421) | 0.1979 (0.1093) | 2.2E-27 |

Query Drift

To see how query drifts, we measure the change in cosine similarity to the true information need over two continuous rounds of feedback and revision. Because the two-stage sampling method contains two rounds of retrieval intrinsically, it is unfair to compare it with the first round of regular relevance feedback when measuring query drift. We thus need to construct an intermediary revised query for the two-stage sampling method after the first round of retrieval.

To construct an intermediary estimate, we take the average of the top K documents including original query. It is easy to see that it is not an unbiased estimate because for terms in q_0 , the top K including original query is not a random sample. But this estimate makes full use of all of the classified documents. Let's call it q_1 . If the two-stage sampling method is able to prevent query drift, the final estimate q_2 should be closer to the true information need than q_1 .

We use the same method to measure estimate quality as in the previous section for both the first and the second round of feedback and revision. For a different estimate, we give the paired t test with the null hypothesis that the average cosine of the second round is equal to that of the first. We also give the percentage of queries whose similarity improves after the second round. For the difference retrieval method, if it improves some queries while degrading others badly, it is not a stable system. We also measure the system stability by the average cosine divided by its standard deviation. Table 3 summarizes the result.

For all four data sets, the final estimate of two-stage sampling method best captures the user's information need. The differences between the first round and second round estimates are statistically significantly different for all methods. A careful examination reveals that for top 10 relevance feedback and cut-off relevance feedback, almost in all cases, the estimate quality decreases on the second round; while for the two-stage estimate, the quality of estimate increases. If we look at the number of queries whose estimate quality increases, we find that two-stage method increases estimate quality for a large portion of the queries. This improvement is not gained in the cost of stability. It reveals that our method generates the most stable estimate. Based on the empirical test, it is fairly clear that two-stage method can better prevent query drift.

CONCLUSION

This study proposes a new query revision technique to better capture the user's information need in information retrieval. A two-stage sampling method is used to generate an unbiased estimate of the user's information need with a smaller covariance matrix. This method is compared with regular relevance feedback with adaptive learning on four testing data sets. The proposed method shows significant improvement on precision-recall. It also successfully prevents query drift which troubles traditional relevance feedback.

Table 3. Estimate Quality and Query Drift*

| Time | First Round (std. Dev.) | Second Round (std. Dev.) | Paired <i>t</i> test | % Better | Stability |
|---------------------------|----------------------------|-----------------------------|-------------------------|----------|-----------|
| Avg. cos(RF10, μ) | 0.475 (0.325) | 0.475 (0.327) | 0.056 | 35% | 1.4545 |
| Avg. cos(RF_CF, μ) | 0.475 (0.318) | 0.475 (0.325) | 0.867 | 45% | 1.4621 |
| Avg. cos(q_2 , μ) | 0.545 (0.224) | 0.576 (0.238) | 0.0000 | 67% | 2.4251 |
| Cranfield | First Round (std. Dev.) | Second Round (std. Dev.) | Paired <i>t</i> test | % Better | Stability |
| Avg. cos(RF10, μ) | 0.4138 (0.1989) | 0.4122 (0.1995) | 0.0000 | 21% | 2.0659 |
| Avg. cos(RF_CF, μ) | 0.4181 (0.1961) | 0.4136 (0.1989) | 0.0000 | 25% | 2.0796 |
| Avg. cos(q_2 , μ) | 0.5308 (0.1989) | 0.5422 (0.1945) | 0.01 | 56% | 2.7881 |
| Medline | First Round (std. Dev.) | Second Round (std. Dev.) | Paired <i>t</i> test | % Better | Stability |
| Avg. cos(RF10, μ) | 0.3276 (0.1291) | 0.3254 (0.1295) | 0.0000 | 10% | 2.5124 |
| Avg. cos(RF_CF, μ) | 0.3353 (0.1273) | 0.3276 (0.1291) | 0.0000 | 17% | 2.5382 |
| Avg. cos(q_2 , μ) | 0.5435 (0.1429) | 0.5974 (0.1464) | 0.0000 | 87% | 4.0801 |
| CISI | First Round (std. Dev.) | Second Round (std. Dev.) | Paired <i>t</i> test | % Better | Stability |
| Avg. cos(RF10, μ) | 0.1924 (0.1100) | 0.1907 (0.1101) | 0.0000 | 12% | 1.7318 |
| Avg. cos(RF_CF, μ) | 0.1979 (0.1093) | 0.1923 (0.1100) | 0.0000 | 13% | 1.7474 |
| Avg. cos(q_2 , μ) | 0.3410 (0.1351) | 0.3544 (0.1241) | 0.01 | 70% | 2.8551 |

*In this table, Avg. cos(RF10, μ) represents the average cosine between top 10 relevance feedback estimate and the user's information need. RF_CF is the estimate for cut-off relevance feedback, and q_2 is the estimate of two-stage sampling. The First Round and Second Round columns have the average cosines and standard deviations for each round respectively.

Since this is a new query revision method, it is interesting to test it in other situations. For example, what will be the converging behavior of the two-stage sampling method? Will it converge to true mean? What will be the result if we test it on a larger data set? If we drop the assumption that terms in a document are independent, then how do we construct an unbiased estimate? These tests will be covered in our future research.

References

- Allan, J. "Incremental Relevance Feedback for Information Filtering," in *Proceedings of SIGIR '96*, Zurich, Switzerland, 1996, pp. 270-278.
- Buckley, C., Salton, G., and Allan, J. "The Effect of Adding Relevance Information in a Relevance Feedback Environment," in *Proceedings of SIGIR'94*, 1994, pp. 292-300.
- Efthimiadis, E. N. "Query Expansion," *Annual Review of Information Systems and Technology (ARIST)* (31), 1996, pp. 121-187.

- Efthimiadis, E., and Biron, P. "UCLA-Okapi at TREC-2: Query Expansion Experiments," in *Proceedings of the Second Test Retrieval Conference (TREC-2)*, NIST Special Publication, 1995.
- Egnor, D., and Lord, R. "Structured Information Retrieval Using XML," in *Proceedings of the SIGIR Work Shop on XML and Information Retrieval*, Springer-Verlag New York, Inc., New York, 2000.
- Fensel, D., Decker, S., Erdmann, M., and Studer, R. "Ontobroker: The Very High Idea," in *Proceedings of the 11th International Flairs Conference (FLAIRS-98)*, Sanibal Island, FL, 1998.
- Hearst, M. A. "Improving Full-Text Precision on Short Queries using Simple Constraints," in *Proceedings of the Symposium on Document Analysis and Information Retrieval*, Las Vegas, NV, April 1996.
- Lieberman, H. "Letizia: An Agent that Assists Web Browsing," in *Proceedings of the International Joint Conference on Artificial Intelligence*, C. S. Mellish (ed.), Morgan Kauffmann Publishers, Inc., San Mateo, CA, 1995.
- Mitra, M., Singhal, A., and Buckley, C. "Improving Automatic Query Expansion," in *Proceedings of SIGIR '98*, August 1998, pp. 11-19.
- Robins, D. "Shifts of Focus in Information Retrieval Interaction," *Journal of the American Society for Information Science*, 2000, pp. 913-928.
- Rocchio, J. J. Jr. "Relevance Feedback in Information Retrieval," Scientific Report ISR-9, Section 23, Harvard Computer Laboratory, Cambridge, MA, August 1965.
- Salton, G., and Buckley, C. "Improving Retrieval Performance by Relevance Feedback," *Journal of the American Society for Information Science* (41:4), 1990, pp. 288-297.
- Salton, G., and Buckley, C. "Optimization of Relevance Feedback Weights," in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, WA, 1995, pp. 351-357.
- Salton, G., and McGill, M. J. *Introduction to Modern Information Retrieval*, McGraw-Hill Book Company, New York, 1983.