Interactive Retrieval using IRIS: TREC-6 Experiments

Robert G. Sumner, Jr., Kiduk Yang, Roger Akers, and W. M. Shaw, Jr.
School of Information and Library Science
University of North Carolina
Chapel Hill, NC 27599-3360 USA
{sumnr, yangk, shaw}@ils.unc.edu
akers@unc.edu

0 Submitted Runs

unc6ia and *unc6ip* – interactive track runs
unc6ma – Category B, manual adhoc task run
unc6aal – Category B, automatic adhoc task run (long query)
unc6aas – Category B, automatic adhoc task run (short query)

1 Introduction

For the TREC-5, Category B adhoc task, we examined the effectiveness of two relevance feedback models: an adaptive linear model and a probabilistic model (Sumner & Shaw, 1997). The models were shown to be effective, especially when the relevance assessments of the searchers matched those of the official TREC judges. During feedback, the query was expanded by a large number of terms from the retrieved documents. Some queries were expanded by as many as 1000 terms.

Building on the basic framework of our TREC-5 system, we developed an interactive, Web-based retrieval system called IRIS (Information Retrieval Interactive System[1]) for TREC-6. Although IRIS inherits both the adaptive linear and the probabilistic model from the TREC-5 system, we made significant modifications to the implementation of both models in order to use a three-valued scale of relevance during feedback. Furthermore, we expanded the scope of human interaction with the system. For example, throughout the search process, the searcher can add and delete query terms as well as change their weights. Moreover, statistically significant, two-word collocations have been added to the term index. IRIS uses collocations not only in formulating the feedback query, but also in presenting to the searcher "suggested phrases" (i.e., collocations related to the initial query), prior to the first document retrieval pass. Finally, as with our TREC-5 system, during feedback the query is expanded by a large number of terms. However, for reasons of efficiency, the number of terms in the query was limited to 300 in our TREC-6 system.

The primary focus of our TREC-6 experiments was on the interactive track and the manual, Category B adhoc task. People were hired to conduct searches for these runs. Here, we are interested not only in the official TREC results but also (perhaps more so) in the reactions of the searchers to the various features of IRIS. The searchers' responses to questionnaires as well as the retrieval effectiveness of the searches are analyzed in this paper as we address, among other things:

- What are the relative effectiveness and the different properties of the adaptive linear and the probabilistic models? Which model do the searchers prefer?

[1] A prior version of IRIS was developed by Kiduk Yang, Kristin Chaffin, Sean Semone, and Lisa Wilcox at the School of Information and Library Science (SILS) at the University of North Carolina. They worked under the supervision of William Shaw and Robert Losee.

- What are the frequencies of documents declared relevant, marginally relevant, and nonrelevant by the searchers? Do searchers utilize all three categories of relevance?
- What is the effectiveness of the suggested collocations? Do searchers find them helpful?

2 Features of IRIS

The features described here apply to the interactive track and manual adhoc runs, and not necessarily to the automatic adhoc runs.

2.1 Stemming and Indexing

The full-text of 210,158 Financial Times (FT) documents was processed to generate a single-word index consisting of 401,423 terms and a collocation index of 400,576 terms. Processing of the full-text involved removing punctuation, numbers, and the 390 high-frequency terms listed in the WAIS default stopwords list. We then conflated morphological variations of words by applying "the modified Krovetz inflectional stemmer."[2]

This stemmer implements a modified version of Krovetz's inflectional stemmer algorithm (Krovetz, 1993). Our stemmer restores the root form of plural ("-s," "-es," "-ies"), past tense ("-ed"), and present participle ("-ing") words, provided this root form is in our online dictionary. The modified Krovetz inflectional stemmer was chosen over other suffix removal stemmers such as Porter's stemmer and SMART's modified-Lovins stemmer, in part due to its conservative approach to stemming. In our TREC-5 experiments (Sumner & Shaw, 1997), we felt that SMART's stemmer incorrectly stemmed too many words and thus had a detrimental effect on precision. For example, "Spence," "Spencer," and "spent" all stemmed to "spent," and "Alger" and "algae" both stemmed to "alg."

2.2 Collocation Index

To augment single-word terms, two-word collocations were automatically extracted from the collection and used to generate a second index. A collocation is defined loosely as a pair of terms that occur together more frequently than normally expected. For descriptive purposes, a collocation consists of a "target word" and its "collocate." Do these frequently co-occurring terms represent a concept or "meaning" which can be automatically extracted and used in information retrieval? "Collocational meaning" is discussed in linguistics and has been investigated for utility in lexicographical tasks (Choueka, Klein, & Neuwitz, 1983; Firth, 1957; Smadja & McKeown, 1990). Our system attempts to take advantage of this collocational meaning to provide a finer level of discrimination between documents.

The collocation indexing process began by extracting from the stemmed collection (without stopwords) all two-word pairs occurring within ±3 words of each other in a paragraph (Haas & Losee, 1994; Losee, 1994; Martin, Al, & van Sterkenburg, 1983; Phillips, 1985). This process generated a very large list of possibly meaningful collocations. It is obvious that using a word window of ±3 words over a collection of documents will result in the extraction of pairs of words that co-occur purely by chance and have no useful syntactic or semantic relationship to each other. Therefore, the next step in creating a useful supplemental index was to cull the list leaving only those two-word pairs co-occurring with outstanding frequency. A z-score representing the probability of the two words co-occurring by chance was calculated for each pair in the list (Berry-Rogghe, 1974). This probability was based on the frequency distribution of the individual terms and the term pairs. All word pairs with a z-score of 2.576 or greater ($\alpha = 0.005$) were considered to co-occur with statistically significant frequency. The final collocation index consisted of statistically significant collocations that occurred more than one time in the collection.

2.3 Ranking Function and Document Term Weights

Documents are ranked in decreasing order of the inner product of document and query vectors,

[2] This stemmer was developed by Kiduk Yang, Danqi Song, Woo-Seob Jeong, and Rong Tang at SILS at UNC.

$$\mathbf{q}^{\mathrm{T}}\mathbf{d}_i = \sum_{k=1}^{t} q_k d_{ik}\,, \qquad (1)$$

where q_k is the weight of term k in the query, d_{ik} is the weight of term k in document i, and t is the number of terms in the index. Document term weights are SMART *Lnu* weights, which were effective in both TREC-4 (Buckley, Singhal, Mitra, & Salton, 1996) and TREC-5 (Buckley, Singhal, & Mitra, 1997). According to Singhal, Buckley, and Mitra (1996), *Lnu* weights were created in an attempt to match the probability of retrieval given a document length with the probability of relevance given that length. Our implementation of *Lnu* weights was the same as that of Buckley et al. (1996, 1997) except for the value of the "slope" in the formula, which is an adjustable parameter. The optimal value for slope may depend, in part, on the properties of the document collection. Based on test runs using TREC-5 topics, we used a slope of 0.3 for the FT collection for both the initial search iteration and feedback iterations. Unfortunately, as explained later, there was a bug in these test runs.

2.4 Initial Query Formulation

Figures 1 through 5 all show different screens in IRIS for the same search. In the search the user is interested in the various drugs used to treat asthma. (This topic was used as an example in NIST's interactive track tutorial for the control system, ZPRISE.)

Figure 1 shows the screen in IRIS where the user enters the initial query for the search. The user can "emphasize" a term in the query by adding an asterisk to the end of it. The user can also enter two-word collocations (or "phrases") in two different ways. To indicate that the query should include, not only the collocation, but its component words as well, the user should enclose the collocation in double quotes. To indicate that the query should not include the component words, the user should enclose the collocation in single quotes.

After the user clicks the "Search" button, IRIS removes stopwords from the query, stems words, and computes SMART *ltc* query term weights (Buckley, C., Salton, G., Allan, J., & Singhal, A., 1995). It also adds 1.0, the maximum possible *ltc* weight, to the *ltc* weights of terms which were emphasized by the user. An "Initial Query Modification" screen is then displayed (see Figure 2). The stemmed term, the number of postings, and the query term weight (multiplied by 10 and rounded off for ease of reading) are listed for each term entered by the user. Terms that are not in the collection's index are not displayed. Since *ltc* weights incorporate inverse document frequency, the weights are inversely proportional to the number of postings. Also, note that "asthma" has a high weight because it was emphasized (see Figure 1). The user can change these query term weights if she wishes. The user can also further modify the query by going back to the previous screen using her Web browser. Alternatively, she can formulate a completely new query by hitting "New Search" at the bottom of the screen.

2.5 Suggested Collocations

At the right of the screen in Figure 2 are collocations "suggested" by IRIS. If the user wishes to add any suggested collocations to her query, she can do so by changing its weight from the preset value of zero. The process by which these "suggested phrases" are chosen by IRIS is now described. The original query posed by the searcher goes through a pre-retrieval process. All two-word collocations found in the original query are extracted following the same extraction procedure used to generate the collocation index. Those collocations from the query considered "significant" in the collection (i.e., those in the collocation index) are placed at the top of the suggested phrase list. In addition, each single query term is used to look up other significant collocations that contain the query term in question. Those collocations are added to the suggested phrase list using the following rules. Collocations that include an emphasized query term are added first to the list. The collocations are then ordered according to the number of query terms with which a collocate pairs. For example, given the initial query in Figure 1, if "tilade" forms a significant collocation with both "asthma" and "drug," then "asthma tilade" and "drug tilade" will be ordered higher on the list than "drug abuse" because "abuse" only collocates significantly with "drug." Finally, the collocations are ordered by decreasing z-score. The top 30 collocations in the list are then presented to the searcher.

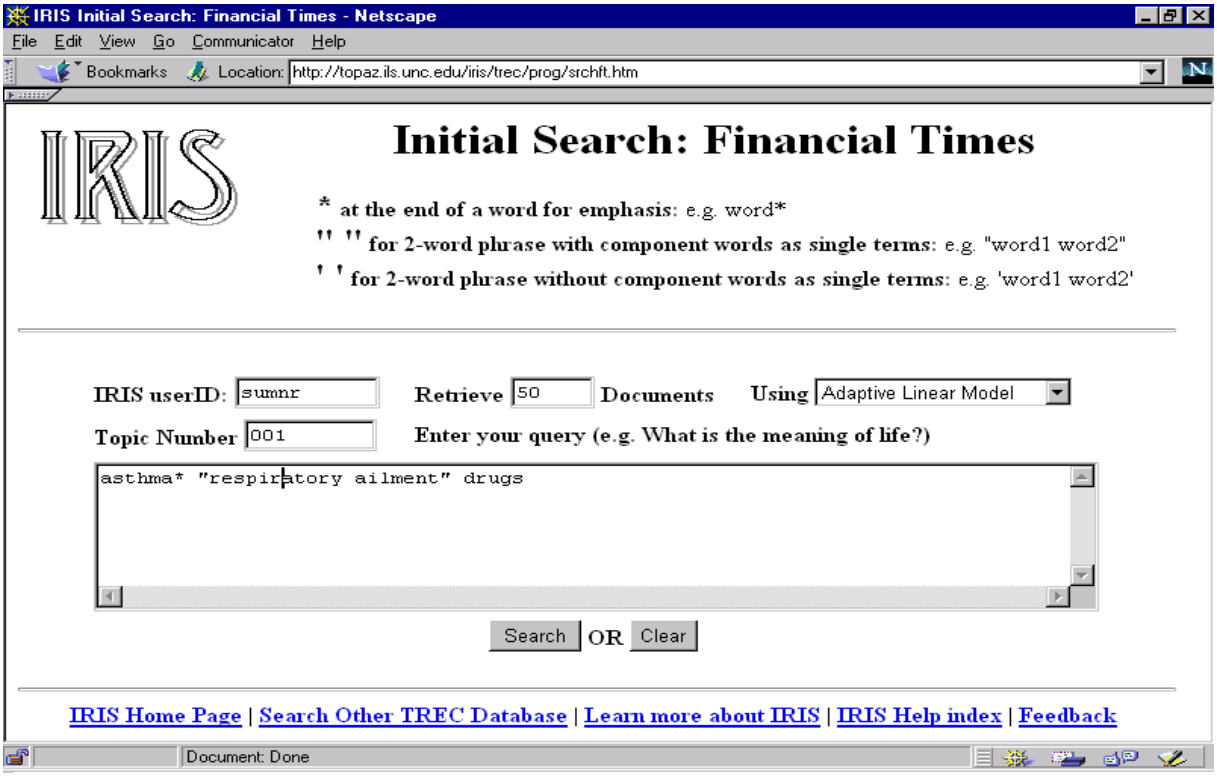

Figure 1: "Initial Search" screen of IRIS.

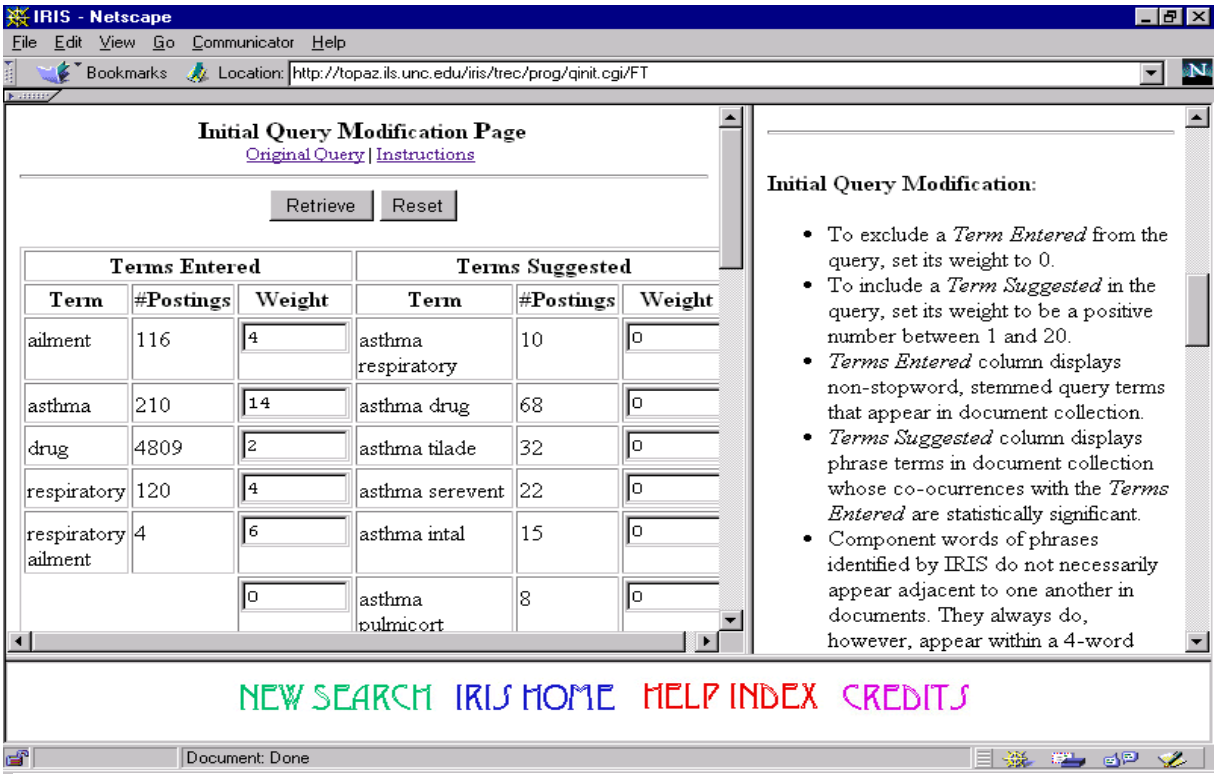

Figure 2: "Initial Query Modification" screen.

2.6 Relevance Feedback

2.6.1 IRIS Features

Figure 3 shows how retrieved documents are displayed in IRIS. Here, the initial ranking of documents is shown. The titles of the retrieved documents are displayed, and they are ranked in decreasing order of the inner product of document and query vectors. If the user clicks on the system-assigned document number to the left of a title, the corresponding document will appear in the frame on the right. The original query terms are boldfaced and the 10 highest-weighted terms in the feedback query vector are italicized in the displayed documents.

The user can utilize relevance feedback to try to improve the search. The user has the option of assigning one of three levels of relevance to a document. In the online instructions for IRIS, it is noted that the "Maybe" category could also be interpreted as "marginally relevant." An additional option is "SAVE," which was added to IRIS specifically for the interactive track. If the user selects "SAVE," the document is designated as "relevant," and it is also added to a system log indicating that the document addresses a new aspect of the query. Finally, the user may forgo the feedback process, if she wishes, by hitting "New Search" or by going back to a previous screen using the Web browser.

There may be cases where only part of a document is relevant to the query, or where a document passage contains words that the user feels should be given high weights in the next iteration of the search. In these cases, the user may wish to use the "Emphasize Terms Box" (see Figure 4). A new browser window is opened and the user can copy and paste into this window a document passage. As with the initial query, the user can indicate the special importance of a term by using an asterisk. She can also signify a collocation using either double or single quotes. Terms added to the query using the Emphasize Terms Box are stemmed and their weights are incremented by the maximum term weight of the feedback query vector. Weights of terms modified by an asterisk are incremented by twice the maximum term weight of the vector.

After making relevance assessments, the user can enter "Resubmit" as shown in Figure 3, and the designated feedback model will produce a query vector consisting of both single-word terms and collocations (see Figure 5). By default, the 25 terms with the highest positive weights and the 25 terms with the lowest negative weights are displayed.

The user can change these weights. Also, the user can add terms to the query. (In the figure, the term "boots" is added because of a pharmaceutical company by that name.) Finally, the user enters "Retrieve" to re-rank the documents.

The query vector produced by the feedback model may contain more than a thousand terms. However, query vectors of this size substantially increase the time it takes to retrieve the documents. Hence, the query vector used in the ranking process was restricted to the 250 terms with the highest positive weights and the 50 terms with the lowest negative weights.

2.6.2 Adaptive Linear Model

One of the relevance feedback models used in our experiments is the adaptive linear model (Bollmann & Wong, 1987; Wong & Yao, 1990; Wong, Yao, & Bollmann, 1988; Wong, Yao, Salton, & Buckley, 1991). This model is based on the preference relation, a concept from decision theory (Fishburn, 1970). Let $\mathbf{D}$ be the set of document vectors for a collection of documents. Then the *user preference relation* $\succ$ on $\mathbf{D}$ is defined as a binary relation on $\mathbf{D}$ where for all $\mathbf{d}_i, \mathbf{d}_j \in \mathbf{D}$,

$$\mathbf{d}_i \succ \mathbf{d}_j \Leftrightarrow \text{the user with a query prefers } \mathbf{d}_i \text{ to } \mathbf{d}_j. \quad (2)$$

An IR model based on the user preference relation allows the use of a multivalued relevance scale such as the three-valued scale used in our TREC-6 experiments. In our TREC-5 experiments (Sumner & Shaw, 1997), the adaptive linear model was used with a binary scale of relevance.

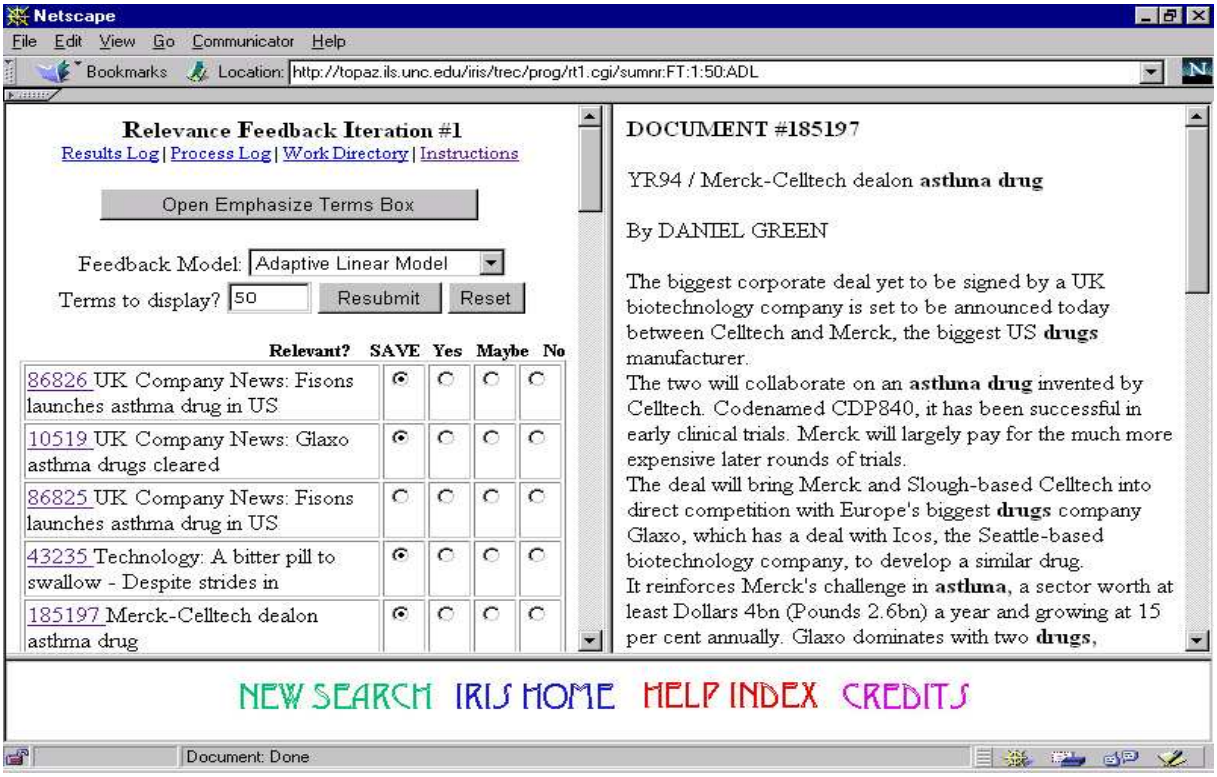

Figure 3: Ranking of documents in IRIS with the text of one of the retrieved documents displayed.

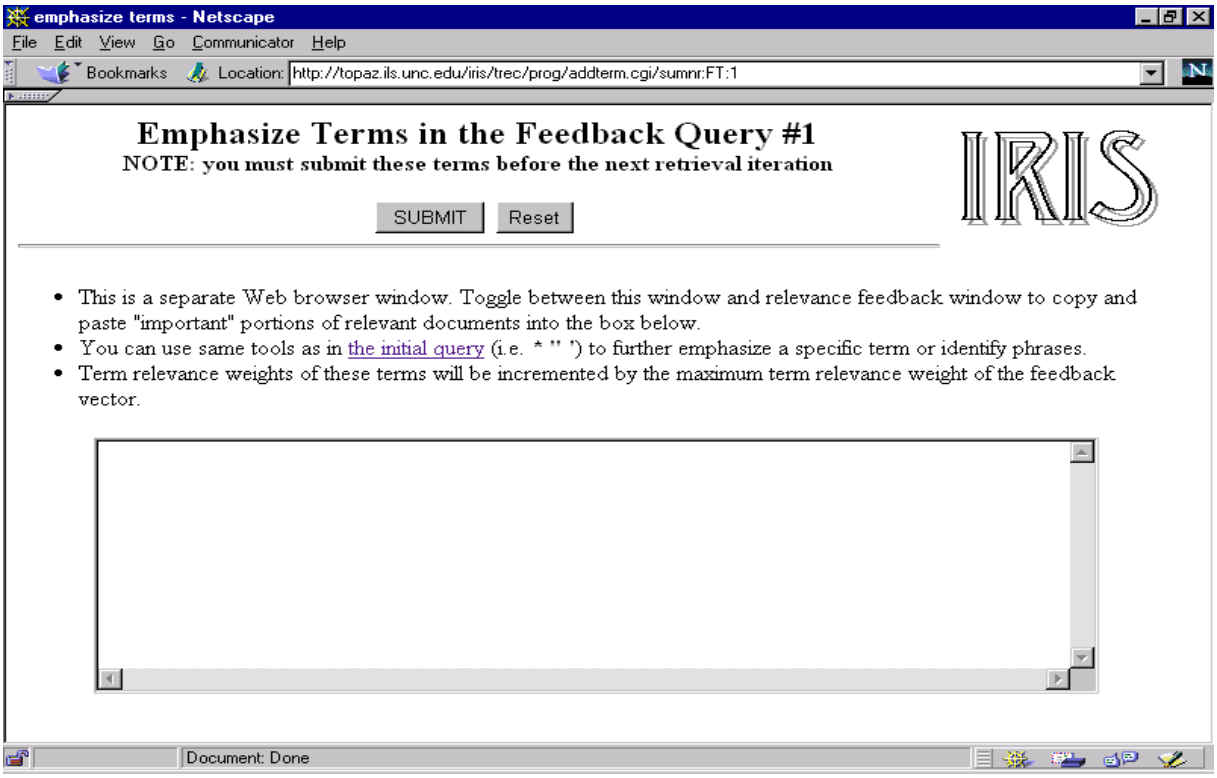

Figure 4: Emphasize Terms Box.

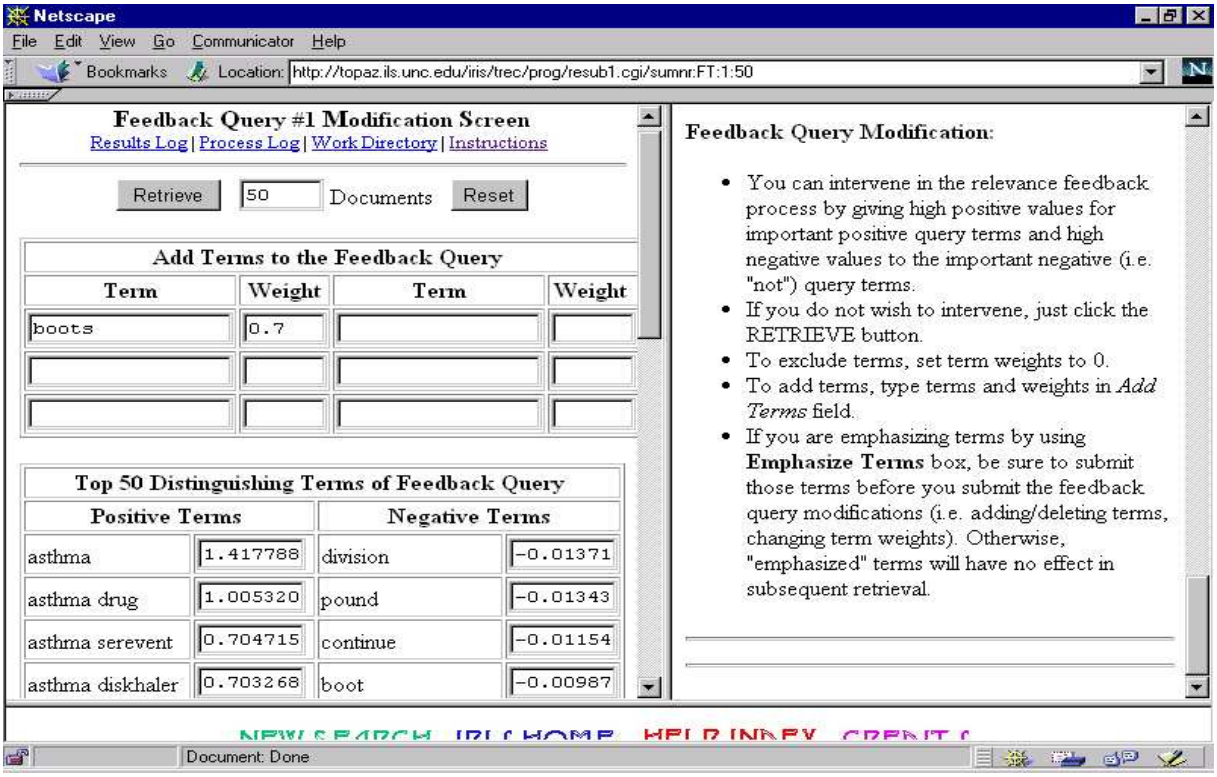

Figure 5: Feedback Query Modification Screen.

The adaptive linear model assumes that the documents in a collection are ranked according to the inner product of document and query vectors. If there exists a query vector $\mathbf{q}$ such that for all $\mathbf{d}_i$, $\mathbf{d}_j \in \mathbf{D}$,

$$\mathbf{d}_i \succ \mathbf{d}_j \Rightarrow \mathbf{q}^T\mathbf{d}_i > \mathbf{q}^T\mathbf{d}_j, \tag{3}$$

then this vector will always rank a more-preferred document before a less-preferred one (Wong et al., 1988). Such a query vector $\mathbf{q}$ is called a *solution vector*, and the set of all solution vectors for $\succ$ on $\mathbf{D}$ are said to comprise a *solution region* in the vector space. A solution vector can be found, provided one exists, by employing an *error-correction procedure* (Nilsson, 1965, Ch. 4; Wong et al., 1988).

However, the user's preferences are not known for the entire collection of documents. They are only known for the *training set*, which consists of those documents that have been retrieved and evaluated by the user up to that point in the search. Accordingly, in the adaptive linear model, a solution vector is found, provided one exists, for $\mathbf{T}$, the set of vectors corresponding to the training set. As $\mathbf{T}$ grows larger, one can expect that the solution region for $\mathbf{T}$ will approach that for $\mathbf{D}$ (Wong & Yao, 1990).

For our TREC-6 experiments, a solution vector for $\mathbf{T}$ was found using a variation of the error-correction procedure used by Wong et al. (1991). During a given cycle i of the algorithm, if query vector $\mathbf{q}_{(i)}$ is a solution vector, the algorithm terminates. If $\mathbf{q}_{(i)}$ is not a solution vector, a new query vector $\mathbf{q}_{(i+1)}$ is created by

$$\mathbf{q}_{(i+1)} = \mathbf{q}_{(i)} + \alpha\mathbf{b}_{max} \tag{4}$$

where α is a positive constant, $\mathbf{B} = \{\mathbf{b} = \mathbf{d}_i - \mathbf{d}_j \mid \mathbf{d}_i, \mathbf{d}_j \in \mathbf{T} \text{ and } \mathbf{d}_i \succ \mathbf{d}_j\}$, and $\mathbf{b}_{max} \in \mathbf{B}$ such that for all $\mathbf{b} \in \mathbf{B}$,

$$-\mathbf{q}_{(i)}^T\mathbf{b}_{max} \geq -\mathbf{q}_{(i)}^T\mathbf{b}. \tag{5}$$

It can be shown that this algorithm will converge to a solution vector if one exists (Nilsson, 1965, pp. 85-87).

Thus, during a given cycle of this algorithm, one document vector is added to the query vector, and another document vector, less relevant than the first one, is subtracted from it. Also, because the desired result is $\mathbf{q}_{(i)}^{T}\mathbf{b} > 0$ for all $\mathbf{b} \in \mathbf{B}$, then the quantity $-\mathbf{q}_{(i)}^{T}\mathbf{b}$ can be viewed as a measure of the extent to which **b** is in error. The vector $\mathbf{b}_{max}$ then is that **b** that produces the maximum error (Wong et al., 1991).

The *starting vector* $\mathbf{q}_{(0)}$ is the initial query vector of the error-correction procedure. The choices made for the starting vector and for the constant α are important because they influence the composition of the solution vector produced by the procedure. These choices may also influence the number of cycles that the procedure runs through before finding a solution vector.

Initially in our research, following Sumner and Shaw (1997), α was 1 and the starting vector was

$$\mathbf{q}_{(0)} = \mathbf{q}_{rk} + \sum_{new\,rel} \mathbf{d}, \tag{6}$$

where $\mathbf{q}_{rk}$ is the query vector that produced the current ranking of documents and where the summation is over all of the *new* relevant documents retrieved. A "new" relevant, retrieved document during a given search iteration is one that was not retrieved and evaluated during a previous iteration. Alternatively, it may also be a document that was declared either "nonrelevant" or "marginally relevant" in a previous iteration, but whose relevance was changed to "relevant" in the current iteration. Sumner and Shaw's choices for α and the starting vector were generalizations in the context of multiple feedback iterations of the choices made by Wong et al. (1991) in the context of one feedback iteration.

We conducted some searches where the interactive functionality of IRIS was tested. Using Equation 6 for the starting vector in these searches, we noticed that documents previously declared as "nonrelevant" were often still near the top of the ranking. We also noticed that documents previously declared as "marginally relevant" were at times "pushed down" a hundred documents or so. Hence, we decided to change the starting vector to *insure* that (1) the vectors of all new nonrelevant documents were subtracted from $\mathbf{q}_{rk}$ and that (2) the vectors of all new marginally relevant documents were added to $\mathbf{q}_{rk}$. (New marginally relevant and new nonrelevant documents are analogous to new relevant documents.) Hence, the following formula was used for the starting vector:

$$\mathbf{q}_{(0)} = c_0\mathbf{q}_{rk} + \frac{c_1}{N_{new\,rel}} \sum_{new\,rel} \mathbf{d} + \frac{c_2}{N_{new\,mrel}} \sum_{new\,mrel} \mathbf{d} - \frac{c_3}{N_{new\,nonrel}} \sum_{new\,nonrel} \mathbf{d}, \tag{7}$$

where c_0, c_1, c_2, and c_3 are constants; $N_{new\,rel}$, $N_{new\,mrel}$, and $N_{new\,nonrel}$ are the number of new relevant, new marginally relevant, and new nonrelevant documents respectively in the current iteration; and the summations, as in Equation 6, are over the appropriate new documents. This formula is similar to the relevance feedback formulas used by Rocchio (1971) and Salton and Buckley (1990). Our formula is adapted for a three-valued relevance scale, though, instead of a binary scale. Of course, these formulas can be generalized to any multi-valued relevance scale.

We used values of $c_0 = 1.0$, $c_1 = 1.2$, $c_2 = 0.6$, and $c_3 = 0.6$ in Equation 7. In addition, we used a value of $\alpha = 0.5$ in Equation 4. Because every vector of a new document is either added to or subtracted from $\mathbf{q}_{rk}$ in Equation 7, we thought that the value for α should be less than one to reduce the influence of any one new document. The value for c_2 is 0.6 so that a new marginally relevant document that is subtracted only one time in the error-correction procedure will contribute some to the final query vector. (In such a situation with an α of 0.5, a marginally relevant document $\mathbf{d}_j$ would contribute $0.1\mathbf{d}_j$ to the final query vector.) Finally, $c_1 = 1.2$ so that the influence of relevant documents would be double that of marginally relevant ones and $c_3 = 0.6$ for internal consistency.

Although there was still the problem of previously declared, nonrelevant documents "floating" to the top of the document ranking, it seemed to be less of a problem using Equation 7 than using Equation 6. Also, marginally relevant documents did not appear to be pushed down as frequently using Equation 7. However, we did not do a systematic investigation of these properties.

Of course, it is possible that there is not a solution vector for $\succ$ on **T**. However, a solution vector usually exists for a set of document vectors like **T**, where the number of vectors in the set are much less than the number of terms in the indexing vocabulary (Nilsson, 1965, pp. 32-35). Wong et al. (1991) and Sumner & Shaw (1997) found solution vectors for every **T** in their experiments—as did Sumner in an unpublished study. Problems can still arise, however, especially in the case where duplicate documents or "near-duplicates" are assigned different levels of relevance.

Searches conducted on the FT collection revealed the presence of either duplicates or near-duplicates. Accordingly, to take into account situations where a solution vector may not exist, the number of cycles in the error-correction procedure was limited to 201, and then $\mathbf{q}_{(201)}$ was returned as the feedback vector. This threshold was also chosen so that the user would not have to wait an inordinately long time for IRIS to produce the feedback vector.

Finally, even though the feedback vector produced by the adaptive linear model may be a solution vector for **T**, the vector actually used to rank the documents in IRIS during the next iteration of the search may not be one. Firstly, the user was allowed to change the weights of terms and also to add terms to the query. Secondly, to increase the speed of the retrieval process, the number of terms in the query vector used to rank the documents was limited to 300. This may mean that a large number of terms are excluded from the query vector. Due to query expansion during the error-correction procedure as well as during the creation of the starting vector, the feedback vector produced by the adaptive linear model may have as many as 1000, or even 5000, terms.

2.6.3 Probabilistic Model

In addition to the adaptive linear model, a variation of the binary probabilistic feedback model used in our TREC-5 experiment (Sumner & Shaw, 1997) was implemented in IRIS. Terms in the feedback query vector came from relevant or marginally relevant documents of the training set. To increase the speed of the retrieval process, the vector was limited to the 250 terms with the highest positive weights and the 50 terms with the lowest negative weights. The traditional binary relevance weight formula (Robertson & Sparck Jones, 1976), however, was modified to accommodate three levels of relevance judgments. Also, *Lnu* document term weights were used by Equation 1 to rank the documents.

The tri-level term relevance weight of term k is denoted by $(tr)_k$ and is defined by

$$(tr)_k = \log\left[\frac{p_k/(1-p_k)}{u_k/(1-u_k)}\right] + \frac{1}{2}\times\log\frac{m_k}{(1-rm_k)}, \tag{8}$$

where p_k is the probability term k appears in a relevant document of the training set, u_k is the probability term k appears in a nonrelevant document of the training set, m_k is the probability term k appears in a marginally relevant document of the training set, and rm_k is the probability term k appears in a relevant or marginally relevant document of the training set. When a term appears in all or none of the relevant, marginally relevant, or nonrelevant documents in the training set, estimations of p_k, u_k, m_k, and rm_k can lead to undefined values of $(tr)_k$ and therefore computing equations must be adjusted to estimate the probabilities in such instances. In TREC-5, we used Shaw's "alternative" computing equation (1995) to determine p_k and u_k instead of the "conventional" 0.5 formula (Robertson & Sparck Jones, 1976), which can overestimate term relevance weights when few relevant documents are detected in the training set (Shaw, 1995; van Rijsbergen, Harper, & Porter, 1981; Yu, Buckley, Lam, & Salton, 1983). Estimation of m_k and rm_k are done in a similar manner:

$$p_k = \frac{r_k}{N_r} \qquad \begin{bmatrix} \frac{1}{N_d^2} & \text{if } r_k = 0 \\ 1-\frac{1}{N_d^2} & \text{if } r_k = N_r \end{bmatrix}, \tag{9}$$

$$u_k = \frac{d_k - r_k}{N_d - N_r} \qquad \begin{bmatrix} \frac{1}{N_d^2} & \text{if } d_k - r_k = 0 \\ 1 - \frac{1}{N_d^2} & \text{if } d_k - r_k = N_d - N_r \end{bmatrix}, \tag{10}$$

$$m_k = \frac{mr_k}{N_{mr}} \qquad \begin{bmatrix} \frac{1}{N_d^2} & \text{if } mr_k = 0 \\ 1 - \frac{1}{N_d^2} & \text{if } mr_k = N_{mr} \end{bmatrix}, \tag{11}$$

$$rm_k = \frac{r_k + mr_k}{N_r + N_{mr}} \qquad \begin{bmatrix} \frac{1}{N_d^2} & \text{if } r_k + mr_k = 0 \\ 1 - \frac{1}{N_d^2} & \text{if } r_k + mr_k = N_r + N_{mr} \end{bmatrix}, \tag{12}$$

where N_d, N_r and N_{mr} are the total number of documents, the total number of relevant documents, and the total number of marginally relevant documents, respectively, in the training set, and d_k, r_k and mr_k are the number of documents, the number of relevant documents, and the number of marginally relevant documents, respectively, in which term k appears. The alternative computing equations with binary relevance judgments have been shown to be highly effective in retrospective and predictive tests in a small retrieval test collection (Shaw, 1995, 1996), and were therefore adapted to three-valued relevance judgments for comparison purposes.

In our TREC-6 experiments, however, we inadvertently discarded the second term of Equation 8. Hence, we used the conventional binary term relevance weight formula.

The tri-level term relevance weight formula (Equation 8), as is the case with the binary term relevance weight formula, is a special case of a more general multi-level relevance formula, which is essentially a document ranking function with graded relevance judgments (Yang & Yang, 1997). It is easy to see that the tri-level term relevance weight formula collapses into the binary term relevance weight when the notion of marginal relevance is taken out. The document ranking function with graded relevance judgments can be shown to preserve the relevance rank order of documents (Yang & Yang); however, the computing formula that estimates the probabilities from the training set remains to be proven. Furthermore, the basic approach of the probabilistic model—i. e., using the training set to estimate the probabilities—risks poor performance when the training set is small, which is often the case in an operational setting.

2.7 Pre-Testing

A number of system decisions with respect to the the manual adhoc task and the interactive track were based, either entirely or in part, on pre-testing using the FT collection and TREC-5 topics. Relevance feedback was simulated automatically using official TREC relevance judgments. Retrieval effectiveness was evaluated using average non-interpolated precision and optimal F values (Shaw, Burgin, & Howell, 1997a, 1997b; van Rijsbergen, 1979).

Several decisions were based on this data. Unfortunately, a bug in these pre-testing runs made their results invalid. First, the adaptive linear model was chosen as the relevance feedback model for the manual adhoc task

over the probabilistic model and a fusion model (Lee, 1995, 1996a, 1996b). Second, a slope of 0.3 was utilized for the *Lnu* document term weights during both the initial search iteration and feedback iterations. Third, collocations were added to the feedback vectors along with single-word terms. Fourth, the query vector was limited to the top 250 positive-weighted terms and the lowest 50 negative-weighted terms. In these test runs with the bug, the best retrieval effectiveness came from the run where the number of terms in the query vector was not limited; however, we decided to limit the number of terms in order to decrease retrieval time. Finally, with respect to the adaptive linear model, the values of $c_0 = 1.0$, $c_1 = 1.2$, $c_2 = 0.6$, and $c_3 = 0.6$ were used for the starting vector in Equation 7.

Even without the bug, it would be difficult to generalize these automatically-generated results to interactive searches on IRIS. First, the official TREC relevance judgments are binary instead of three-valued. Second, in IRIS users can add terms to the feedback vector, delete terms, and change their weights. Third, it is difficult to simulate other aspects of the retrieval behavior of users such as the number of documents that are examined during a given search iteration. The great variation in searching behavior among users makes this task especially daunting.

3 Interactive Track Runs

3.1 Methodology

We submitted two interactive track runs: *unc6ia* and *unc6ip*. The adaptive linear model was employed in *unc6ia* and the probabilistic model in *unc6ip*. The four searchers for *unc6ia* were designated as *irisa1i* through *irisa4i*, and the four searchers for *unc6ip* were designated as *irisp5i* through *irisp8i*. See the Appendix for information about the searchers.

Each searcher conducted her interactive track searches during one 3 ½ to 5 hour session. She first filled out a "Pre-Study Questionnaire," from which information was gathered about her background and searching experience. She then read the "introductory instructions" from the Interactive Track Specification (Over, 1997a). She next proceeded to search on one system (either IRIS or ZPRISE) and then the other. For each system, the same sequence of events occurred. First, one of us trained the searcher on the system. An attempt was made to standardize the training, but there may have been some (mostly minor) differences between one training session and another. Second, the searcher conducted a practice search as if it were a real interactive track search. Each person searched on the same practice topic, depending on whether the system in question was the first system for that person or the second. Third, the searcher was given feedback on her practice search. Fourth, she conducted the official interactive track searches. We had suggested that she write down on a "Searcher Worksheet" at the beginning of the search, aspects that she thought may exist about a topic, and that it may be worthwhile to use the words describing these aspects in her initial query. Once she found and saved a document that covered this aspect she was to put a checkmark next to the aspect on the worksheet. Likewise, if she came across a document covering an aspect that she had not previously thought of, she was to write some words describing it on the worksheet and then put a checkmark next to those words. After each search, she filled out a "Post-Search Questionnaire," and after all three searches on the system, she filled out a "Post-System Questionnaire." Finally, after searching on both systems, she filled out an "Exit Questionnaire."

In departure from the Interactive Track Specification (Over, 1997a), we either told or implied to the searchers that they should spend the full 20 minutes on each topic. First, we did this because of our experience with the manual adhoc searches, which usually took at least thirty minutes and sometimes as long as an hour. These searches took a long time because of the searchers' thoroughness and because IRIS and the Web client-server architecture can be slow at times. It often took one or more minutes for documents to be retrieved and displayed. Second, we did it because we thought it would be easier to implement than allowing the time per search to vary. However, in hindsight, we probably should have allowed the searcher to spend no more time on a search than she wanted to. Then our searches would be more comparable to those of the other interactive track participants.

During our training session, we also gave some "hints" to searchers about how best to conduct the searches. For example, for IRIS searches, we suggested that after they save a document that covers a specific aspect of the topic, they give a high negative weight to the term in the feedback vector that best describes that aspect. Then the feedback vector will probably not retrieve documents that cover that specific aspect but may retrieve documents that cover other aspects. However, as it turned out, few searchers implemented this suggestion.

3.2 Results

Table 1 has our TREC-6 interactive track results with respect to aspectual recall, and Table 2 has the results with respect to aspectual precision. In the tables, E refers to the Experimental system (IRIS), and C refers to the Control system (ZPRISE). See Over (1997a) for a detailed explanation of the interactive track evaluation measures including how the "control-adjusted response" (E–C) is estimated.

Tables 1 and 2 include the mean of the six estimates of E–C for aspectual recall (aspectual precision), the standard deviation of the six estimates, the 95% confidence interval for mean E-C, the mean of the aspectual recall (aspectual precision) values for the twelve searches using E, and the mean of the aspectual recall (aspectual precision) values for the twelve searches using C. The two values given for standard deviation and the confidence interval correspond to two different ways that E-C can be estimated. For some of the measures, the run's rank with respect to the nine other interactive track runs is given.

Table 1: Aspectual recall results. Rank among 10 interactive track runs is given in parentheses.

Run	Mean of Six E–C Estimates	Standard Deviation E–C*	95% Confidence Interval for Mean E–C*	Mean E Recall	Mean C Recall
unc6ia	-0.067 (7)	0.108 0.129	-0.181 to 0.046 -0.202 to 0.068	0.444 (6)	0.511 (1)
unc6ip	0.012 (5)	0.119 0.081	-0.113 to 0.136 -0.073 to 0.096	0.467 (4)	0.455 (5)

*The top value corresponds to the "sum.out" estimate (Over, 1997b) and the bottom value corresponds to the "sum-alt.out" estimate (Over, 1997c).

Table 2: Aspectual Precision Results. Rank among 10 interactive track runs is given in parentheses.

Run	Mean of Six E–C Estimates	Standard Deviation E–C*	95% Confidence Interval for Mean E–C*	Mean E Precision	Mean C Precision
unc6ia	-0.154 (10)	0.231 0.222	-0.396 to 0.089 -0.387 to 0.079	0.595 (10)	0.749 (8)
unc6ip	0.013 (3)	0.305 0.343	-0.307 to 0.333 -0.348 to 0.373	0.785 (5)	0.772 (6)

*The top value corresponds to the "sum.out" estimate (Over, 1997b) and the bottom value corresponds to the "sum-alt.out" estimate (Over, 1997c).

There were several problems with our searches that have a bearing on our results. First, searcher *irisa2i* did not save any documents during her three searches on IRIS. She put checkmarks next to aspects on her Searcher Worksheet though. We think she probably thought that marking a document as "Relevant" would be the same as saving it (see Figure 3). Hence, for her searches, any documents that she marked as relevant, we later marked as saved. Second, our system logs had no record of any evaluated documents (saved, relevant, etc.) for IRIS search *4_347* in run *unc6ia* and for IRIS search *6_307* in run *unc6ip*. However, on the Searcher Worksheets, 10 aspects were marked as saved for *4_347*, and 8 aspects were marked as saved for *6_307*. It is still unclear to us how these evaluated documents were lost. Third, with the exception of *irisp7i*, we neglected to tell the searchers that they should hit "Resubmit" in IRIS if they were viewing a ranking of documents at the end of the 20 minute time limit (see Figure 3). If they did not hit "Resubmit," any new documents that they had saved during that iteration would not be logged as saved by IRIS. Through examination of time logs, we conjecture that this may have had a negative impact on at least 1 search in *unc6ia* and at least 5 searches in *unc6ip*. (This problem may also have been a factor in searches *4_347* and *6_307*.) Finally, for search *1_326* in *unc6ia*, a bug in IRIS adversely affected the final ranking of documents.

In addition, two searchers (*irisa2i* and *irisa3i*) wrote on their Searcher Worksheets words from their queries instead of words used to describe aspects of the topic. For example, for Topic 339i the different aspects of the topic were the various drugs used to treat Alzheimer's Disease, but *irisa2i* wrote on her worksheet the words "alzheimer," "drug," "success," "treatment," "pharmaceutical," and "glaxo." The first four of these words had checkmarks next to them. An examination suggests that these searchers may have put checkmarks next to terms that had retrieved documents that they then saved. More investigation is needed to determine to what extent *irisa2i* and *irisa3i* misunderstood the goals of the interactive track experiment. Their values for aspectual recall are, in general, not worse than those for other UNC searchers.

3.3 Discussion

In the interactive track, mean E–C values are the principal measures used to determine whether differences exist among the experimental systems with respect to aspectual recall and aspectual precision. However, a high degree of overlap among the 95% confidence intervals for the ten runs makes any differences in the mean E–C values less meaningful. With respect to both aspectual recall and aspectual precision, the confidence interval for the best run overlaps the confidence interval for each of the nine other runs when either method for determining the confidence intervals is used (Over, 1997b, 1997c).

Regarding the performance of our two experimental systems, it is difficult to make a definitive statement because of the problems outlined earlier concerning the logging of saved documents. In addition, instructing searchers that they should utilize the full 20 minutes when searching on a topic may have also had a negative impact on our results. ZPRISE is faster than IRIS at retrieving documents, so, over the same time period, more search iterations can be conducted using ZPRISE. This may explain, in part, why run *unc6ia* had the best aspectual recall for ZPRISE out of the ten interactive track runs (see Table 1).

For run *unc6ia*, the searchers found that the collocations suggested by IRIS for possible addition to the initial query were helpful for most of the searches (see Table 3). Their responses are mixed for *unc6ip* (see Table 4).[3] As described previously, IRIS added to its term index those collocations that it determined to be statistically significant. It appears that at least one of the two words for most of these collocations occurred in a small number of documents. Hence, the collocations suggested by IRIS are perhaps most helpful with respect to topics that have specific aspects that are covered by few documents. For example, for the search on drugs used to treat asthma that was described previously, there were 10 suggested collocations that included the word "asthma" and the name of a drug used to treat it. However, a drawback of the method used to determine the statistically significant collocations is that many useful collocations that are not infrequent are not added to the term index. Accordingly, many collocations added to the query by searchers may not be in our index. This shortcoming was recognized too late in the process to correct in time for our TREC-6 runs.

Again, *unc6ia* employed the adaptive linear model, and *unc6ip* employed the probabilistic model. The searchers in *unc6ia* found relevance feedback in IRIS to be more beneficial to their sessions than the searchers in *unc6ip* (see Tables 3 and 4). In addition, on the exit questionnaires, searchers were asked which system they "liked the best" between IRIS and ZPRISE. The four searchers in *unc6ia* said "IRIS," whereas, in *unc6ip*, only one said "IRIS," while two said "ZPRISE" and one said she could not decide. It is difficult to generalize from two sample sets of only four searchers each. However, it is possible that the two relevance feedback models have different properties which may have influenced, in part, the searchers' responses.[4] Further investigation of the properties of the two models is needed.

[3] The last two questions in Tables 3 and 4 were taken directly from the "Post-Search Questionnaire" used by Rutgers (Belkin et al.) in the TREC-6 interactive track pre-experiment.

[4] However, any properties of the feedback models would not explain the differences between the two runs with respect to the searchers' attitudes toward the suggested collocations. Also, it is unclear how much of an influence on the responses was the fact that, for *unc6ip*, the marginally relevant documents were essentially treated like nonrelevant documents because the second term of Equation 8 was inadvertently ignored. Although further examination is needed, this may have had a relatively minor influence on the searchers' responses.

Table 3: For *unc6ia*, frequencies of answers to those questions on the IRIS Post-Search Questionnaire concerning the searcher's perceptions of the results of a search and of the impact on it by the suggested collocations and relevance feedback.

To what extent…	**Not at all** 1	2	**Marginally** 3	4	**Extremely** 5
were the suggested phrases for the initial iteration of the search helpful?		2		8	2
did relevance feedback help retrieve documents that cover new aspects?	2	1	3	5	1
did relevance feedback contribute in a positive way to the search?	2		2	7	1
did relevance feedback contribute in a negative way to the search?*	4	3	3	1	
are you satisfied with your search results?	2	3	3	3	1
are you confident that you identified all the possible aspects for this topic?	6	3		3	

*No answer was given for one search.

Table 4: For *unc6ip*, frequencies of answers to those questions on the IRIS Post-Search Questionnaire concerning the searcher's perceptions of the results of a search and of the impact on it by the suggested collocations and relevance feedback.

To what extent…	**Not at all** 1	2	**Marginally** 3	4	**Extremely** 5
were the suggested phrases for the initial iteration of the search helpful?	3	2	2	4	1
did relevance feedback help retrieve documents that cover new aspects?	4	3	4	1	
did relevance feedback contribute in a positive way to the search?	5	1	5	1	
did relevance feedback contribute in a negative way to the search?	2	7	2	1	
are you satisfied with your search results?	2	1	3	5	1
are you confident that you identified all the possible aspects for this topic?	6	2	3	1	

Although the searchers who used the adaptive linear model seemed to find relevance feedback more helpful than the searchers who used the probabilistic model, run *unc6ip* had better results in the interactive track than run *unc6ia* with respect to mean E–C values for both aspectual recall and aspectual precision. However, because the 95% confidence intervals overlap, the difference between the models could be explained by chance. In any case, further investigation of the relative retrieval effectiveness of the models is needed.

We were also interested in the number of documents declared relevant, marginally relevant, and nonrelevant by the searchers. These frequencies were also calculated for our manual adhoc run, so both sets of numbers will be presented in Section 5.

4 Manual Adhoc Runs

Run *unc6ma* was our Category B, manual adhoc run. Seven searchers searched for us using IRIS and the adaptive linear model. The searchers were either currently or had recently been graduate students in Library Science or

Information Science at UNC. Three searchers did 9 topics each, three did 6 topics each, and one did 5 topics. Most, if not all, of the searches took at least thirty minutes, and some took as long as an hour.

Table 5 contains performance values for the run, which are averaged over the 47 topics with at least one relevant FT document. In addition to overall average non-interpolated precision, the table includes average precision for the top 10, 20, and 30 documents retrieved. The last three values are included because we feel that (1) retrieval performance should be high for that set of documents that the typical searcher will evaluate and that (2) the typical searcher will usually not examine more than the top 30 documents. Finally, it should be noted that examining the final ranking of documents may not be the optimal way to evaluate an interactive retrieval session with multiple search iterations. The searcher may use one iteration of the search to retrieve documents that cover one aspect of the topic, and may use another iteration to retrieve documents that cover a different aspect. Our searchers, however, tried to produce the best final ranking of documents that they could.

Table 5: Performance measures for the manual, adhoc run (*unc6ma)*. Values are averaged over the 47 topics with at least one relevant FT document.

Average non-interpolated precision	0.3663
Precision at 10 documents	0.4277
Precision at 20 documents	0.3309
Precision at 30 documents	0.2794

There were a few bugs in the system due to the fact that we were rushing to meet the TREC deadline. We are only aware of two topics that were affected by bugs. However, an in-depth investigation of the effect of bugs on our results is needed. One bug adversely affected the results for Topic 321. Another bug *only* affected Topic 303. The searcher appears to have been able to overcome the effects of this bug during the later iterations of the search.

After they had completed all of their searches, the searchers filled out an exit questionnaire. The frequencies of their answers to some of its questions are given in Table 6.[5] As in the interactive track, the searchers in general found that the suggested collocations were helpful. However, they did not think relevance feedback was as helpful as the searchers did for the interactive run *unc6ia* which also used the adaptive linear model. In fact, as shown in Table 6, a number of the manual searchers claimed that feedback contributed to the failure of searches. Oral and written comments by the searchers may explain, in part, the reason for this. Their comments indicate that there is still the problem of previously declared, nonrelevant documents floating to the top of the ranking. This is not a problem when the probabilistic model is used. This problem is particularly frustrating when the searcher wants to create a final query that will place all of the relevant documents before all of the nonrelevant ones. A simple solution to this problem though is to not include these previously declared, nonrelevant documents in the displayed ranking. (Not including them could be the "default" option in IRIS.)

5 Frequencies of Evaluated Documents

For both the interactive track and the manual adhoc task, we are interested in the number of documents declared relevant, marginally relevant, and nonrelevant by the searchers. Figures 6 and 7 display this information for the interactive track runs, *unc6ia* and *unc6ip*, respectively. The number of iterations in a search are displayed as well as the searcher's answer on the post-system questionnaire concerning the number of relevance levels that the searcher preferred to use.[6] For each searcher, the data are displayed in the order in which the topics were searched. (Again, the log of evaluated documents was lost for the search on topic 347i by *irisa4i* and for the search on topic 307i by *irisp6i*.) In each figure, the two searchers on the left searched on IRIS before they searched on ZPRISE, and the

[5] It should be stressed again that, in Tables 3 and 4 which refer to the interactive track searches, the questionnaires were filled out after *each search* on IRIS. In contrast, in Table 6, the questionnaire was filled out after all of the searcher's searches were completed.

[6] This question appears to have been misinterpreted by *irisa2i*, so her answer is not shown. Also, the options given the searchers were two levels, three levels, and four levels or higher (they could fill in a number). We did not think to include one level of relevance as an option (i.e., the only level would be "relevant").

two searchers on the right searched on ZPRISE before IRIS. The figures do not include any documents that were not logged due to the searcher's failure to hit "Resubmit."

Table 6: For *unc6ma*, frequencies of answers to those questions on the exit questionnaire concerning the searcher's perceptions of the results of the search and of the impact on it by the suggested collocations and relevance feedback. The exit questionnaire was filled out after all of the searcher's searchers were completed.

Rate the following…	**Not at all** 1	2	**Marginally** 3	4	**Extremely** 5
How confident are you that search outcomes were successful?			4	3	
Were the suggested phrases for the initial iteration of the search helpful?		1	3	1	2
Did relevance feedback contribute to the success of searches?			6	1	
Did relevance feedback contribute to the failure of searches?		1	4	2	
Were you satisfied with using three levels of relevance?	2		3	2	

Several points can be made about Figures 6 and 7. First, there is a high degree of variation among searchers with respect to the total number of documents evaluated as well as the percentage of documents assigned a given level of relevance. Second, for most of the searches, a high percentage of documents were declared nonrelevant. Third, only seven out of the twenty-two searches had documents declared marginally relevant.[7] Finally, an order effect can be detected. The searchers who searched on ZPRISE before IRIS generally seem to have evaluated fewer documents than those who searched on IRIS before ZPRISE. This difference is perhaps due to fatigue because the searches took place during one 3 ½ to 5 hour session. There is also some evidence of an order effect among the topics searched. In no case did the third topic searched have the most evaluated documents and in several cases it had the fewest.

Figures 8 through 10 show the number of documents declared relevant, marginally relevant, and nonrelevant by the manual adhoc searchers.[8] There are some similarities between Figures 8-10 for the adhoc task and Figures 6-7 for the interactive track even though the nature of the retrieval task was different between the two sets of runs. First, like the interactive track data, the adhoc task data show a high degree of searcher variation with respect to the total number of documents evaluated as well as the percentage of documents assigned a given level of relevance. (In addition, the adhoc task data show a high degree of searcher variation with respect to the number of iterations searched.) These results suggest that an operational IR system incorporating feedback needs to take into account such variation. Second, like the interactive track searches, many of the adhoc searches had a high percentage of documents declared nonrelevant. If this finding is substantiated by further research in a non-laboratory setting using users with real information needs, it would suggest that feedback retrieval systems perhaps should include a "nonrelevant" option as well as a "relevant" one.

[7]With respect to the *unc6ip* run, we need to further investigate whether discarding the second term in Equation 8 had an influence on the number of documents declared marginally relevant.

[8] The manual adhoc searchers indicated the number of relevance levels they preferred to use on the exit questionnaire. Also, like the interactive track searchers, the manual adhoc searchers may have hit "New Search" (see Figure 3) to restart a search using a new query. However, because the goal of the adhoc task was to produce a final query, the data shown in the figures are only for that sequence of iterations (possibly after "New Search" was entered) that produced that final query. Limiting the data in this way also makes the figures less cluttered with information about the number of iterations searched.

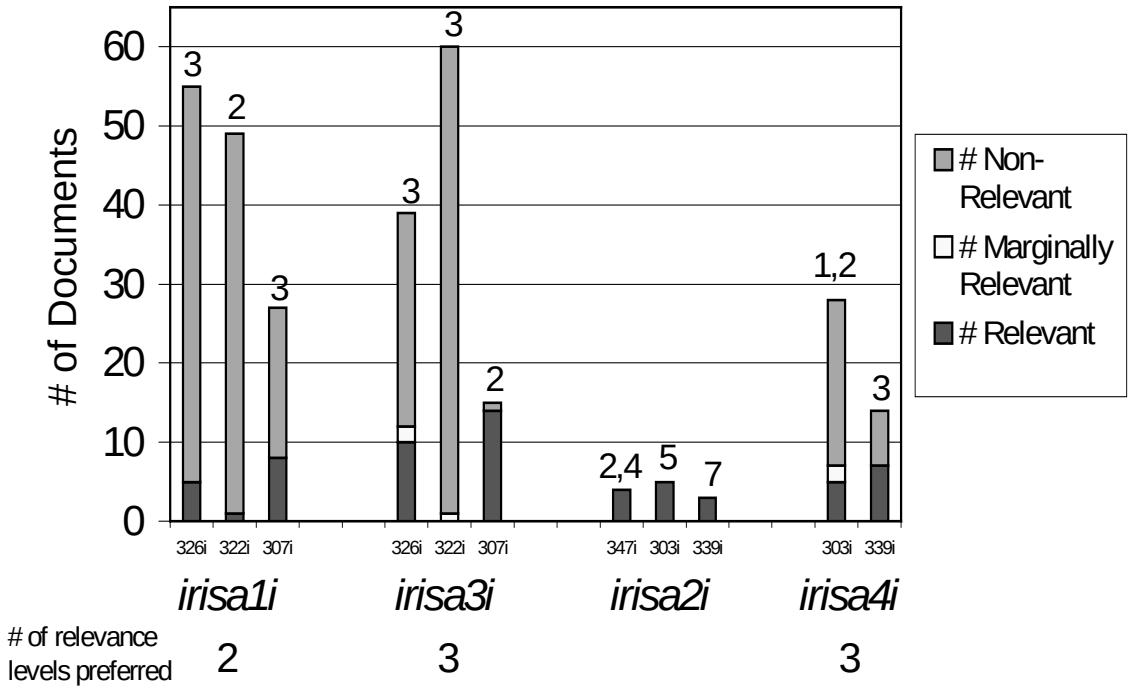

Figure 6: Number of documents declared relevant, marginally relevant, and nonrelevant for a topic by searchers in run *unc6ia*. The number of iterations in a search is given above the appropriate column. (Two numbers are given if the searcher hit "New Search" to restart the search with a fresh query—see Figure 3.)

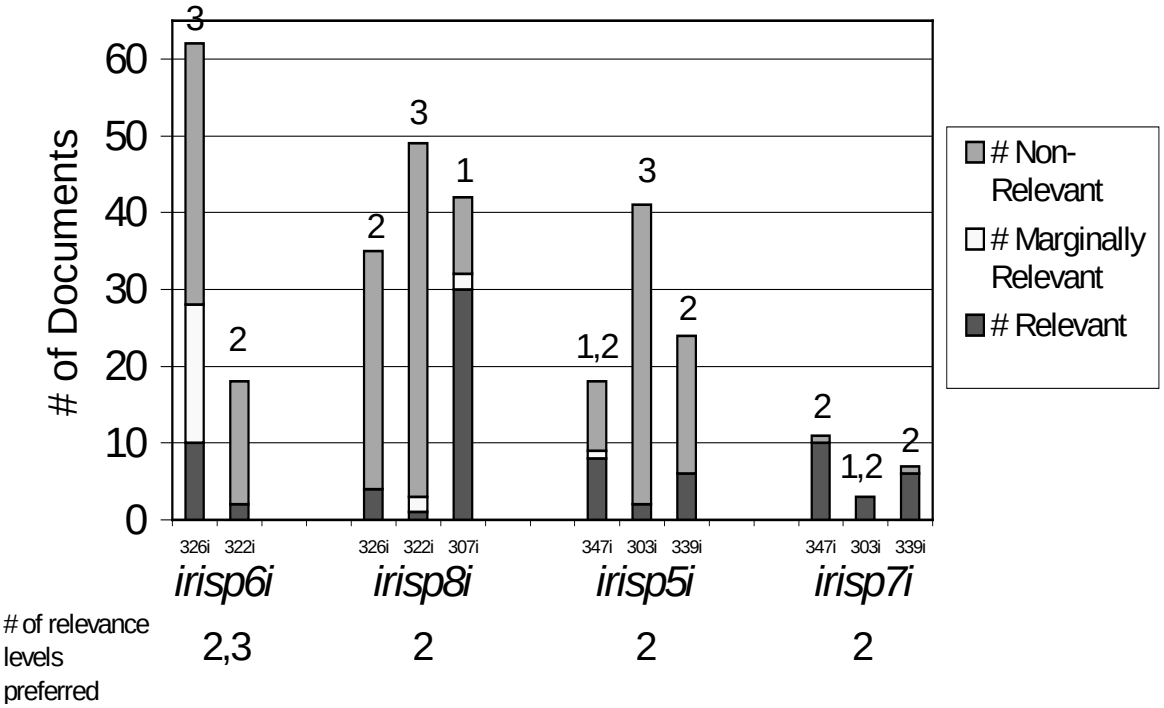

Figure 7: Number of documents declared relevant, marginally relevant, and nonrelevant for a topic by searchers in run *unc6ia*. The number of iterations in a search is given above the appropriate column. (Two numbers are given if the searcher hit "New Search" to restart the search with a fresh query—see Figure 3.)

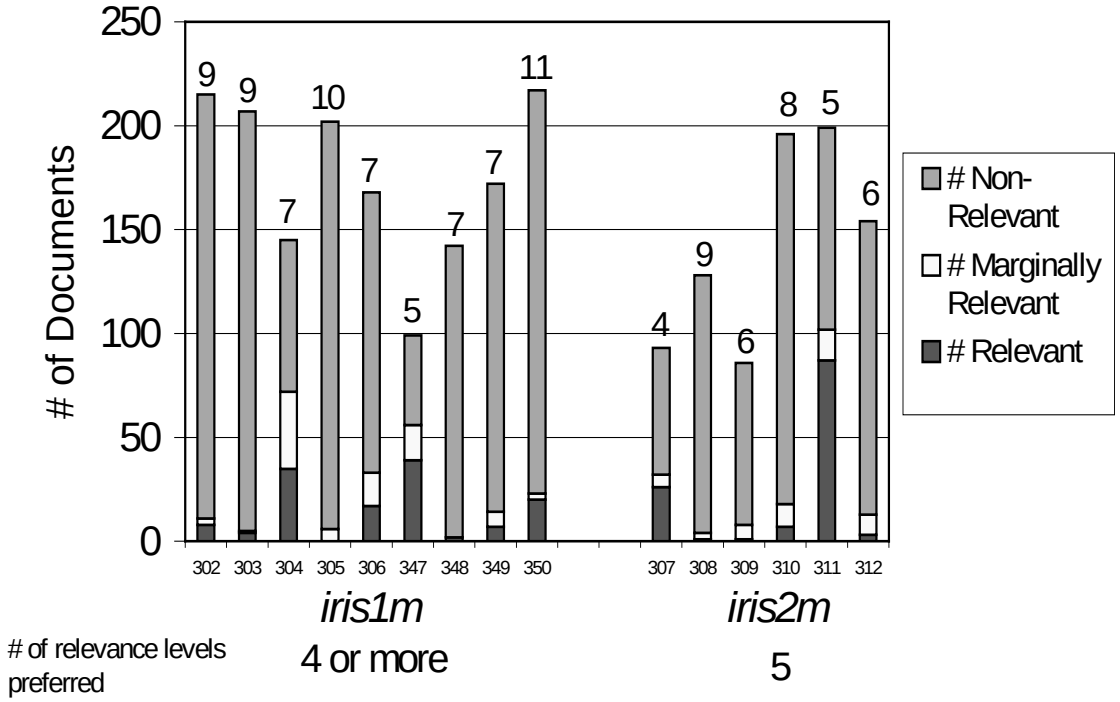

Figure 8: Number of documents declared relevant, marginally relevant, and nonrelevant for a topic by searchers *iris1m* and *iris2m* in run *unc6ma*. The number of iterations in a search is given above the appropriate column.

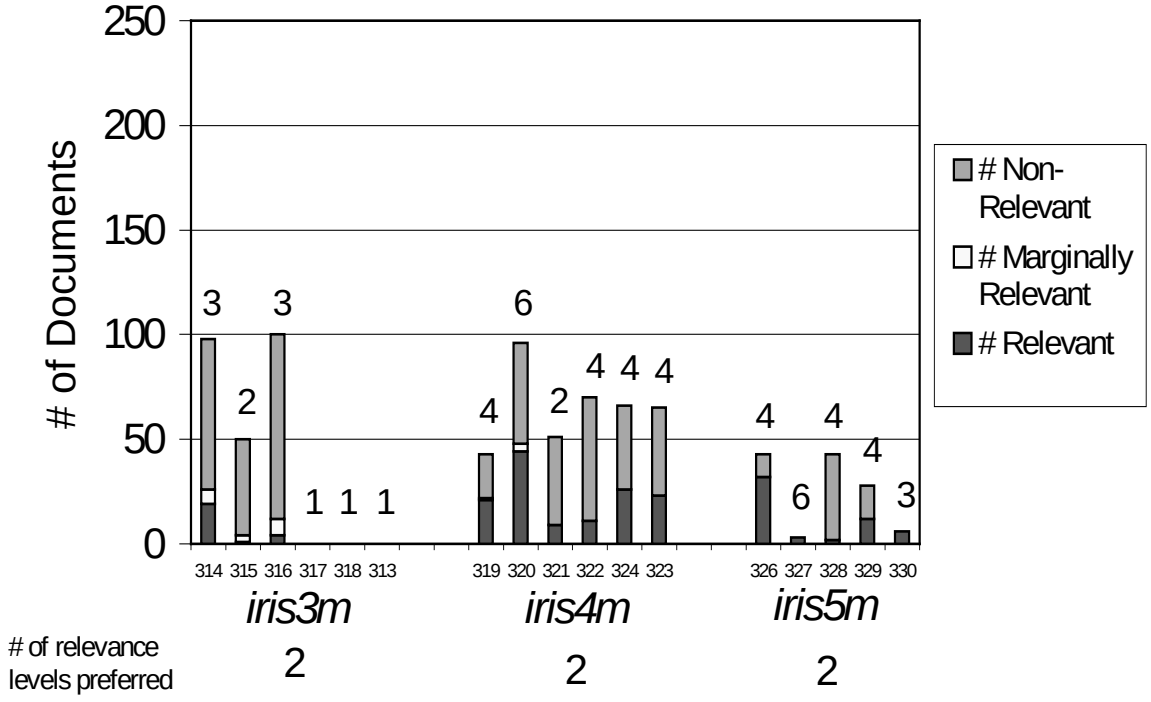

Figure 9: Number of documents declared relevant, marginally relevant, and nonrelevant for a topic by searchers *iris3m*, *iris4m*, and *iris5m* in run *unc6ma*. The number of iterations in a search is given above the appropriate column.

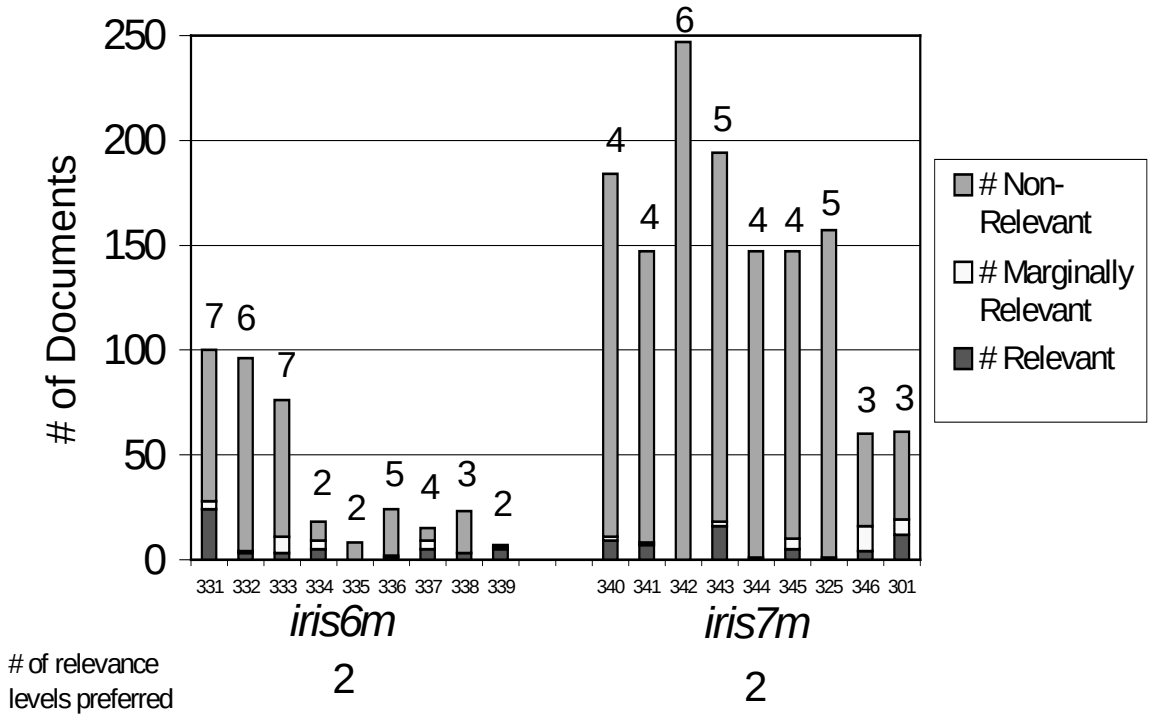

Figure 10: Number of documents declared relevant, marginally relevant, and nonrelevant for a topic by searchers *iris6m* and *iris7m* in run *unc6ma*. The number of iterations in a search is given above the appropriate column.

There are also some differences between the figures for the adhoc task and the figures for the interactive track. First, the adhoc searchers did not have a time limit and sometimes took more than an hour per search. Hence, in contrast to the interactive track searchers, the adhoc searchers generally evaluated a larger number of documents and conducted a greater number of search iterations. Second, a larger number of the adhoc searches had documents that were declared marginally relevant. Thirty-two out of the 47 topics with evaluated documents had at least one document declared marginally relevant. If one excludes the searches of *iris4m* and *iris5m*, this ratio increases to 30 out of 36. This difference between the adhoc task and the interactive track is perhaps explained by the different nature of the two retrieval tasks as well as the greater amount of time spent by the adhoc searchers. Clearly, more research is needed to determine if feedback retrieval systems should include more than two levels of relevance. Our data has other evidence as well (besides the number of searches) concerning whether or not another level of relevance should be included. On the one hand, the figures for the adhoc task show that the number of marginally relevant documents for a search is often a high percentage of the number of relevant documents. On the other hand, a large majority of our TREC searchers said they preferred a binary relevance scale.

5 Automatic Adhoc Runs

Two Category B, automatic adhoc runs were submitted. Run *unc6aas* was the "short query" run, and run *unc6aal* was a "long query" run which utilized the description and narrative fields of the topic. The features of IRIS that do not require any human interaction with the system were employed in the automatic run. Accordingly, the initial query only consisted of single-word terms. No "suggested" collocations were added to it because that requires human judgment concerning which collocations would be the appropriate ones to add. The features that do not require human interaction were implemented the same way that they were in the interactive track and the manual adhoc task (e.g., the document term weights were Lnu weights with a slope of 0.3.)

Participants in previous TREC conferences have explored *top-document* feedback, where the top *X* documents as ranked by the original query are assumed to be relevant and then a feedback model produces a new query vector to re-rank the documents (e.g., Buckley et al., 1995). Of course, the success of this procedure is dependent on the quality of the initial query (Harman, 1996). We investigated using top-document feedback in which the adaptive linear and the probabilistic model are employed. We also tested using more than two iterations in this process. For example, let us assume that three iterations in all are utilized (one of which is the initial iteration). First, the top *X* documents from the initial ranking are assumed to be relevant and are used to produced a new query vector which re-ranks the documents. Then the top *X* documents from the second ranking are assumed to be relevant and are added to the training set in order to produce the final query vector which, in turn, produces the final ranking of documents. If the quality of the initial ranking is poor, using more than two iterations should have an adverse effect on retrieval performance.

We conducted our tests on a subset of the TREC-5 topics, and evaluated the results using the FT relevance judgments. The top *X* documents were assumed to be relevant and the next $100 - X$ were assumed to be nonrelevant. We varied the number of iterations and the "window size" (the value for *X*) in our tests. Table 7 has the results for the long query (title, description, and narrative) and for the adaptive linear and the probabilistic model. Firstly, the adaptive linear model performed much better than the probabilistic model, probably because nonrelevant documents are generally given lower ranks by the probabilistic model as opposed to the adaptive linear model. Accordingly, many of the $100 - X$ documents that are "officially" relevant will be given low ranks by the probabilistic model. Secondly, the window size of 5 performed marginally better than larger window sizes, perhaps due to the fact that, in general, the density of officially relevant documents is probably highest in that window. Thirdly, an unexpected result was that in some cases three iterations did better than two. Finally, the best result was for the adaptive linear model with a window size of 5 and with two iterations. These were the parameters that were used in the "long query" automatic run for TREC-6. However, it should be noted that the best result is only marginally better than the result for the initial iteration.

Table 7: Overall average non-interpolated precision for the long query.

Number of Iterations	Window Size					
	Adaptive Linear Model			Probabilistic Model		
	5	20	30	5	20	30
1	0.2082	0.2082	0.2082	0.2082	0.2082	0.2082
2	**0.2181**	0.2109	0.2060	0.0985	0.0836	0.0647
3	0.2176	0.2150	0.2097	0.1380	0.0463	0.0350

Table 8 contains the results of our testing using the 31 TREC-5 topics and the short query. Only the adaptive linear model was tested because of its superior performance using the long query. Results are similar to that for Table 7. Again, the best run was the adaptive linear model with a window size of 5 and with two iterations. These parameters were used in our TREC-6 short query run.

Table 8: Overall average non-interpolated precision for the short query and the adaptive linear model.

Number of Iterations	Window Size	
	5	20
1	0.1911	0.1911
2	**0.2020**	0.1905
3	0.2019	0.1846

Table 9 contains our official TREC-6 results for both *unc6aas* (the short query run) and *unc6aal* (the long query run). As to be expected, the manual run (see Table 5) did much better than the automatic runs, and the long query automatic run did better than the short query automatic run.

Table 9: Performance measures for the automatic adhoc task for both *unc6aas* (the short query run) and *unc6aal* (the long query run). Values are averaged over the 47 topics with at least one relevant FT document.

	unc6aas	*unc6aal*
Average non-interpolated precision	0.2167	0.2518
Precision at 10 documents	0.2766	0.3064
Precision at 20 documents	0.2138	0.2340
Precision at 30 documents	0.1738	0.1972

7 Future Research

We plan on improving IRIS over the next year or so. First, we may explore modifications to our method for determining statistically significant collocations. Other phrase generation methods may also be investigated. Second, we need to determine the number of levels of relevance with which the user evaluates documents for feedback. Third, we may explore other relevance feedback models that incorporate multiple levels of relevance. Fourth, we may compare using different starting vectors and values for α in the adaptive linear model (see Equation 4). Fifth, we may give the user more control over the feedback process by requiring her to explicitly add the new terms suggested by feedback. This is a model employed by Belkin et al. (1998) and others. Sixth, we may explore presenting the feedback terms to the user after each relevance evaluation of a document (Beaulieu & Gatford, 1998; Belkin et al., 1998) instead of waiting for the user to hit "Resubmit" after she has evaluated a number of documents. Seventh, we are currently working on an online interactive tutorial. Eighth, we are also working on ways to improve our interface. Ninth, we need to make IRIS faster. Finally, the most important thing we need to do is more testing with users with real information needs.

Acknowledgements – Chris Brannon and Scott Barker have given us invaluable computing support. Also, we would not have been able to do the interactive track and the manual adhoc task without our enthusiastic searchers: Sai Balu, Danielle Borasky, David Borasky, Linda Brett, Sally Fessler, Lisa Greenbaum, Wanda Gunther, Anne Langley, Karl Lietzan, Muzhgan Nazarova, Mark Rosso, Robin Shapiro, Lisa Smith, Rong Tang, and Lucinda Thompson. Finally, we would like to belatedly thank Judd Knott and Scott Barker for their indispensable computing support for our TREC-5 experiments.

References

Beaulieu, M. M., Gatford, M. J.. (1998) Interactive Okapi at TREC-6. *Proceedings of the Sixth Text REtrieval Conference*.

Belkin, N. J., Perez-Carballo, J., Cool, C., Lin, S., Park, S. Y., Rieh, S. Y., Savage, P., Sikora, C., Xie, H., & Allan, J. (1998). Rutgers' TREC-6 interactive track experience. *Proceedings of the Sixth Text REtrieval Conference*.

Berry-Rogghe, G. (1974). The computation of collocations and their relevance in lexical studies. In A. J. Aitken, R. W. Bailey, & N. Hamilton-Smith (Eds.), *The Computer and Literary Studies* (pp. 103-112). Edinburgh: Edinburgh University Press.

Bollmann, P., & Wong, S. K. M. (1987). Adaptive linear information retrieval models. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 157-163.

Buckley, C., Salton, G., Allan, J., & Singhal, A. (1995). Automatic query expansion using SMART: TREC 3. In D. K. Harman (Ed.), *Overview of the Third Text REtrieval Conference (TREC-3)* (NIST Spec. Publ. 500-225, pp. 69-80). Washington, DC: U.S. Government Printing Office.

Buckley, C., Singhal, A., & Mitra, M. (1997). Using query zoning and correlation within SMART: TREC 5. In E. M. Voorhees & D. K. Harman (Eds.), *The Fifth Text REtrieval Conference (TREC-5)*.

Buckley, C., Singhal, A., Mitra, M., & Salton, G. (1996). New retrieval approaches using SMART: TREC 4. In D. K. Harman (Ed.), *The Fourth Text REtrieval Conference (TREC-4)* (NIST Spec. Publ. 500-236, pp. 25-48). Washington, DC: U.S. Government Printing Office.

Choueka, Y., Klein, S. T., & Neuwitz, E. (1983). Automatic retrieval of frequent idiomatic and collocational expressions in a large corpus. *Association for Literary and Linguistic Computing Journal*, 4(1), 34-38.

Firth, J. R. (1957). Modes of meaning. *Papers in Linguistics 1934-1951* (pp. 190-215). London: Oxford University Press.

Fishburn, P. C. (1970). *Utility theory for decision making*. New York: John Wiley & Sons.

Haas, S. W., & Losee, R. M. (1994). Looking into text windows: Their size and composition. *Information Processing and Management, 30*, 619-629.

Harman, D. (1996). Overview of the Fourth Text REtrieval Conference (TREC-4). In D. K. Harman (Ed.), *The Fourth Text REtrieval Conference (TREC-4)* (NIST Spec. Publ. 500-236, pp. 25-48). Washington, DC: U.S. Government Printing Office.

Krovetz, R. (1993). *Viewing morphology as an inference process.* Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 191-203.

Lee, J. H. (1995). Combining multiple evidence from different properties of weighting schemes. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 180-188.

Lee, J. H. (1996a). *Analyses of multiple evidence combination (Tech. Rep. No. IR-88).* Amherst: University of Massachusetts, Center for Intelligent Information Retrieval.

Lee, J. H. (1996b). *Combining multiple evidence from different relevance feedback methods (Tech. Rep. No. IR-87).* Amherst: University of Massachusetts, Center for Intelligent Information Retrieval.

Losee, R. M. (1994). Term dependence: Truncating the Bahadur Lazarsfeld expansion. *Information Processing and Management, 30*, 293-303.

Martin, W., Al, B., & van Sterkenburg, P. (1983). On the processing of a text corpus. In R. Hartmann (Ed.), *Lexicography: Principles and practice* (pp. 77-87). London: Academic Press, Inc.

Nilsson, N. J. (1965). *Learning machines: Foundations of trainable pattern-classifying systems*. New York: McGraw-Hill.

Over, P. (1997a). *TREC-6 interactive track specification*. At http://www-nlpir.nist.gov/~over/t6i/trec6spec. (Also see: "TREC-6 Interactive Track Report" in *Proceedings of the Sixth Text REtrieval Conference* (1998).)

Over, P. (1997b). *sum.out*. At http://www-nlpir.nist.gov/~over/t6i/sum.out. (Also see: "TREC-6 Interactive Track Report" in *Proceedings of the Sixth Text REtrieval Conference* (1998).)

Over, P. (1997c). *sum-alt.out*. At http://www-nlpir.nist.gov/~over/t6i/sum-alt.out. (Also see: "TREC-6 Interactive Track Report" in *Proceedings of the Sixth Text REtrieval Conference* (1998).)

Phillips, M. (1985). *Aspects of text structure*. Amsterdam: Elsevier Science Publishers.

Robertson, S. E., & Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science, 27*, 129-146.

Rocchio, J. J., Jr. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART Retrieval System: Experiments in Automatic Document Processing* (pp. 313-323). Englewood Cliffs, NJ: Prentice-Hall.

Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science, 41*, 288-297.

Shaw, W. M., Jr. (1995). Term-relevance computations and perfect retrieval performance. *Information Processing & Management*, 31(4), 491-498.

Shaw, W. M., Jr. (1996). [Letter to the editor]. *Information Processing & Management*, 32, 636-637.

Shaw, W. M., Jr., Burgin, R., & Howell, P. (1997a). Performance standards and evaluations in IR test collections: Cluster-based retrieval models. *Information Processing & Management, 33*, 1-14.

Shaw, W. M., Jr., Burgin, R., & Howell, P. (1997b). Performance standards and evaluations in IR test collections: Vector-space and other retrieval models. *Information Processing & Management, 33*, 15-36.

Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document length normalization. *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 21-29.

Smadja, F., & McKeown, K. (1990). Automatically extracting and representing collocations for language generation. *Proceedings of the 28th Annual Meeting of the Association of Computational Linguistics*, pp. 252-259.

Sumner, R. G., Jr., & Shaw, W. M., Jr. (1997). An investigation of relevance feedback using adaptive linear and probabilistic models. In E. M. Voorhees & D. K. Harman (Eds.), *The Fifth Text REtrieval Conference (TREC-5)*.

van Rijsbergen, C. J. (1979). *Information retrieval* (2nd ed.). London: Butterworths.

van Rijsbergen, C. J., Harper, D. J., & Porter, M. F. (1981). The Selection of Good Search Terms, *Information Processing and Management, 17*, 77-91.

Wong, S. K. M., & Yao, Y. Y. (1990). Query formulation in linear retrieval models. *Journal of the American Society for Information Science, 41*, 334-341.

Wong, S. K. M., Yao, Y. Y., & Bollmann, P. (1988). Linear structure in information retrieval. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 219-232.

Wong, S. K. M., Yao, Y. Y., Salton, G., & Buckley, C. (1991). Evaluation of an adaptive linear model. *Journal of the American Society for Information Science, 42*, 723-730.

Yu, C. T., Buckley, C., Lam, K., & Salton, G. (1983). A generalized term dependence model in information retrieval. *Information Technology: Research and Development, 2*, 129-154.

Yang, K., & Yang, K. (1997). Graded relevance in information retrieval. *Unpublished manuscript.*

Appendix: Characteristics of Interactive Track Searchers

Information about each searcher's background and searching experience was gathered from the pre-study questionnaires. All eight searchers had received at least a bachelor's degree. With respect to the *unc6ia* run, all four searchers had received a Master's in Library Science, were currently working as librarians, and were female. The number of years they had been "online searching" was 9, 2, ~15, and 3. With respect to the *unc6ip* run, one searcher was currently working on a Master's in Information Science, and another was working on hers in Library Science. A third searcher was a library technical assistant, and another was an administrative assistant for the University. Two of the searchers for the *unc6ip* run were female, and two were male. The number of years they had been online searching was 2, 2, 21, and 2. For both *unc6ia* and *unc6ip*, Tables A-1 and A-2 respectively show the frequencies of searchers' answers to questions regarding their searching experience. Many of these questions were directly taken from the "Pre-Search Questionnaire" used by Rutgers (Belkin et al.) in the TREC-6 interactive track pre-experiment.

Table A-1: For *unc6ia*, frequencies of searchers' answers to questions on the Pre-Study Questionnaire regarding searching experience.

How much experience have you had…	**None** 1	2	**Some** 3	4	**A great deal** 5
searching on computerized library catalogs					4
searching on CD ROM systems, e.g., Infotrac, Grolier			2	2	
searching on commercial online systems, e.g., Dialog, Lexis, BRS Afterdark			3	1	
searching on world wide web browsers, e.g., Mosaic, Netscape, Internet Explorer				1	3
searching on other systems	1			1*	
searching full-text databases		2	1	1	
searching in ranked-output information retrieval systems	2		1		1
searching in information retrieval systems that provide relevance feedback	3		1		
using a mouse-based interface				1	2
reading articles from the Financial Times	4				
reading articles from another business- or financial-oriented newpaper, magazine, or other publication (e.g., *The Wall Street Journal*, *BusinessWeek*)		2	1	1	

*OCLC, WorldCat

Table A-2: For *unc6ip*, frequencies of answers to questions on the Pre-Study Questionnaire regarding searching experience.

How much experience have you had…	**None** 1	2	**Some** 3	4	**A great deal** 5
searching on computerized library catalogs			1	2	1
searching on CD ROM systems, e.g., Infotrac, Grolier		1	3		
searching on commercial online systems, e.g., Dialog, Lexis, BRS Afterdark	1	1		1	1
searching on world wide web browsers, e.g., Mosaic, Netscape, Internet Explorer			1	1	2
searching on other systems	1		1*	1**	
searching full-text databases		1	2	1	
searching in ranked-output information retrieval systems	1	2	1		
searching in information retrieval systems that provide relevance feedback	1	2	1		
using a mouse-based interface	1			1	2
reading articles from the Financial Times	4				
reading articles from another business- or financial-oriented newpaper, magazine, or other publication (e.g., *The Wall Street Journal*, *BusinessWeek*)		1	2	1	

*library card catalogs, yellow pages, phone directory
**MEDLINE, UNCLE, OVID