Evaluating Relevance Feedback Algorithms for Searching on Small Displays

Vishwa Vinay[1], Ingemar J. Cox[1],
Natasa Milic-Frayling[2], and Ken Wood[2]

[1] Department of Computer Science,
University College London, UK
v.vinay@cs.ucl.ac.uk, ingemar@ieee.org
[2] Microsoft Research Ltd,
7 J.J.Thomson Avenue, Cambridge, UK
{natasamf, krw}@microsoft.com

Abstract. Searching online information resources using mobile devices is affected by displays on which only a small fraction of the set of ranked documents can be displayed. In this paper, we ask whether the search effort can be reduced, on average, by user feedback indicating a single most relevant document in each display. For small display sizes and limited user actions, we are able to construct a tree representing all possible outcomes. Examination of the tree permits us to compute an upper limit on relevance feedback performance. Three standard feedback algorithms are considered - Rocchio, Robertson/Sparck-Jones and a Bayesian algorithm. Two display strategies are considered, one based on maximizing the immediate information gain and the other on most likely documents. Our results bring out the strengths and weaknesses of the algorithms, and the need for exploratory display strategies with conservative feedback algorithms.

1 Introduction

The continuing evolution of portable computing and communications devices (cell phones, PDAs, etc.) means that more and more people are accessing information and services on the Internet with devices that have small displays. This small display size presents challenges. The need for extensive scrolling makes viewing of standard pages very difficult. Also, devices like mobile phones still lack the resources needed to perform sophisticated processing on the client side.

We are particularly concerned with implications that small display devices have on searching online information resources. Generally, it has been observed that users engage in a variety of information seeking tasks, from "finding" a specific, well defined piece of information, to "gathering information" as a more open ended, research oriented activity ([21]). Adoption of Internet enabled mobile phones is still in its infancy and no general patterns of use have been established. Anticipating that mobile users will search for specific, well defined information, we study the methods which will enable the users to perform the operations of searching for a target.

D.E. Losada and J.M. Fernández-Luna (Eds.): ECIR 2005, LNCS 3408, pp. 185 – 199, 2005.

In this study we explore the effectiveness of relevance feedback methods in assisting the user to access a predefined target document through searching or browsing. We devise an innovative approach to study this problem by exploiting the fact that the display size and thus the user's choices are limited. It is then feasible to generate and study the complete space of a user's interactions and obtain the upper bound on the effectiveness of relevance feedback. This bound represents the actions of an "ideal user" who at every step makes choices that enable the system to reach the target in the minimum number of iterations.

We believe that analysis of the complete search space is a novel experimental paradigm and can lead to interesting insights into the behavior of relevance feedback algorithms. This approach has the further advantage of permitting the study of relevance feedback and display strategies without the need for time-consuming user studies. This, in turn, allows a far greater number of experiments to be performed and we are optimistic that the statistical evidence gathered in this way can be used to predict actual user performance. This will be verified in future work.

In Section 2 we give an overview of the related research for mobile devices and relevance feedback and describe the particular algorithms we use here. In Section 3 we describe the display strategies that we consider - (i) one that maximizes the likelihood that the target is in the display (Top-K) and (ii) one that maximizes the immediate information gain. Experimental results characterize these two strategies. In Section 4 we describe the experimental procedure. In Section 5 we present the results and conclude with a summary of the presented work and an outline of the future research directions.

2 Background

A considerable body of research has been dedicated to the issues related to user interaction ([10][11]), browsing ([1][2]), searching ([21][23]), and reading ([4]) on mobile devices, and the idea of using relevance feedback or other adaptive measures to aid searching on small devices is not new. Most directly relevant to our study is Toogle [22], a front end application that post-processes Google results based on the user's actions, i.e., the user's clicks on documents in the ranked list. Toogle collects evidence, i.e., relevant and non-relevant documents from a single or multiple screens of search results, and applies machine learning techniques to re-rank the remaining documents.

In contrast, our approach focuses on searching using mobile devices and constrains the user feedback model to selection of a single relevant document at each iteration. Under these conditions, we take advantage of the small display size and limited space of user actions to study the full interaction space and all possible outcomes determined by the relevance feedback and display strategies. We are therefore able to provide an upper bound on the performance of relevance feedback systems for small displays.

2.1 Relevance Feedback

Conceptually, a system that involves user relevance feedback can be described by a three-phase iterative process as depicted in Figure 1. This three phase process can represent most, if not all, relevance feedback algorithms.

During the display phase, typically manifested as a list, the user is presented with a number of documents and given an opportunity to indicate which documents are relevant and which are not. This information is then used by the relevance feedback algorithm to induce a new ranking of documents in the database. The new ranking is the basis of the system's next display of a new set of documents to the user. And the process repeats. The process may begin with an initial query to the ranking engine, as depicted, or by a display of some selection of documents generated by the system itself. A good overview of relevance feedback techniques can be found in [8].

In our case, the display phase is the presentation of four documents from the ranked list. The user feedback phase is a single action where the user nominates one of the four displayed documents as most relevant to his or her information need. The document ranking phase applies one of three relevance feedback algorithms, described below, to create a new query based on a weighted combination of the previous query and terms from document selected by the user, and this new query is then used to compute the next ranking of the document collection.

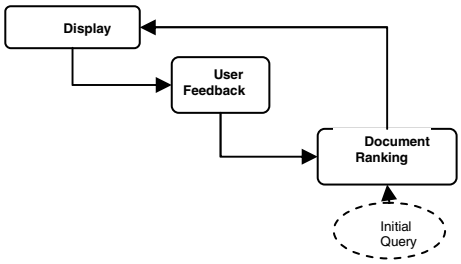

Fig. 1. Relevance Feedback

2.2 The Rocchio Algorithm

The Rocchio relevance feedback scheme [17] is used in conjunction with the term-frequency inverse-document-frequency (tf-idf) representation where documents and queries are represented as vectors of term weights and similarity is measured by the cosine distance between these vectors.

A document is a vector $\boldsymbol{d_i}=(d_{i,1},d_{i,2},\ldots,d_{iT})$ where T is the number of words across the collection, excluding a predefined set of stopwords, and $d_{i,j}= t(i,j)\cdot s_j$. Here $t(i,j)$ corresponds to the number of occurrences of term j in document i and s_j is the inverse document frequency of term j across the whole collection. A query $\boldsymbol{q}= (q_1,q_2,\ldots,q_T)$ is defined similarly, though their values are typically 0 or 1. Both documents and queries are normalized for length by setting

$$d' = \frac{d}{\|d\|} \text{ and } q' = \frac{q}{\|q\|} \text{ where } \|x\| = \sqrt{\sum_{j=1}^{t} x_j^2}$$

and the similarity score between document $\boldsymbol{d}$ and query $\boldsymbol{q}$ is then given by the dot product of the normalized vectors, i.e., $score_{rocchio}(\boldsymbol{d_i},\boldsymbol{q}) = \boldsymbol{d_i}'\cdot\boldsymbol{q}'$. The Rocchio algorithm takes a set **R** of relevant documents and a set **N** of non-relevant documents (as

selected in the user feedback phase) and updates the query weights according to the following equation:

$$w'_j = \alpha\ w_j + \beta \frac{\sum_{i \in \mathbf{R}} d_{i,j}}{n_{\mathbf{R}}} + \gamma \frac{\sum_{i \in \mathbf{N}} d_{i,j}}{n_{\mathbf{N}}}$$

where

n_R and n_N are the number of relevant and non-relevant documents respectively.

The parameters α, β, and γ control the relative effect of the original weights, the relevant documents, and the non-relevant documents. We do not have non-relevant documents and we use $\alpha = \beta = 1$.

2.3 The Robertson/Sparck-Jones Algorithm

In the Robertson/Sparck Jones model of information retrieval [19], the terms in a corpus are all assigned relevance weights which are updated for a particular query whenever relevant documents are identified. Initially the relevance weights are given idf-based values. Documents are given ranking scores against a query based on the relevance weights of the query terms occurring in each document. We use the following formulation of this model. The initial relevance weight for term j is given by

$$w_j = log\ (C / n_j\ )$$

where C is the total number of documents in the corpus and n_j is the number of documents containing term j.

A document $\mathbf{d}_i$ is assigned a score against query **q** as follows:

$$score_{rsj}(\mathrm{d_i}, \mathrm{q}) = \sum_{j \in Q} \frac{(K+1) * t(i,j)}{K(1-b) + \frac{b * |d_i|}{l} + t(i,j)}$$

where

t(i,j) is the number of occurrences of term j in document $\mathbf{d}_i$
K and b are parameters typically set to 2.0 and 0.75 respectively
$|\mathbf{d}_i|$ is the length of document $\mathbf{d}_i$
l is the average length of all documents in the corpus

Documents are then ranked in descending score order. If certain documents are flagged as relevant, the relevance weights are updated as follows:

$$w_j = \log\left(\left(\frac{(r_j + 0.5)}{(n_j - r_j + 0.5)}\right)\left(\frac{(C - n_j - n_R + r_j + 0.5)}{(n_R - r_j + 0.5)}\right)\right)$$

where

R is the number of relevant documents
r_j is the number of relevant documents containing term j
C and n_j are defined as before

In addition to updating the relevance weights, the relevant documents are used to select new (or additional) query terms according to the offer weights, o_j , where $o_j = r * w_j$

Terms are ranked in decreasing order of offer weight, and the top terms are used as part of the subsequent query. How many such terms are to be chosen per iteration is another parameter of the system.

2.4 The Bayesian Algorithm

The Bayesian relevance feedback algorithm [5], first proposed for a Content-Based Image Retrieval System – PicHunter – is a recursive probabilistic formulation in which, at each iteration, k, the probability, P_k of document $\mathbf{d}_i$, being the target document, $\mathbf{d}_T$, is computed. This probability is conditioned on all current and past user actions and the history of displayed documents, which collectively is denoted by H_k . The concept of a current query, **q**, is not explicitly present in this formulation. Thus, at each iteration, the document rankings are given by

$$score_{bayesian}(d_i) = P_k(d_i = d_T \mid H_k)$$
$$= P_{k-1}(d_i = d_T \mid H_{k-1}) * G(d_i, R))$$

where

P_{k-1} is the document's probability in the previous iteration

R is the set of documents marked relevant in this iteration

$G(\mathbf{d}_i, R)$ is given by

$$G(d_i, R) = \prod_{j \in R} \left(\frac{\exp\left(\frac{sim(d_i, d_j)}{\sigma} \right)}{\left(\sum_{((k \in D) and\ (k \notin R))} \exp\left(\frac{sim(d_i, d_k)}{\sigma} \right) \right) + \exp\left(\frac{sim(d_i, d_j)}{\sigma} \right)} \right)$$

The term *sim(x,y)* computes the similarity of document x with document y, which for textual documents can be taken as the cosine dot product of *tf-idf* vectors normalized for length. σ is a tuning noise parameter which is set according to the specific dataset.

3 Display Strategies

At each search iteration, it is necessary to display K documents to the user. The most obvious strategy is to display the K documents with the highest rank. This *Top-K display* is likely to result in a set of documents all very similar to one another. If these documents are close to the target (or even include it), then this may well be optimum. However, if the target is not similar to any of the documents in the currently displayed set, then it is very difficult for a user to direct the search away from the displayed documents and towards the target. This problem has been previously discussed in the context of content-based image retrieval [5] and observed in the current experiments (see Section 6.1.1 – on Convergence). An alternative approach is to display docu-

ments for which a user's response would be most informative to the system and help minimize the number of search iterations. This was proposed in [5] and formulated as finding a selection of K documents that maximizes the *immediate information gain* from the user's response in each iteration. Unfortunately, determining such a document selection is computationally expensive. However, it can be approximated by sampling K documents from the underlying similarity score distribution. There are computationally efficient methods for performing this sampling - usually, this is done by simulating a roulette wheel with the size of each item's field proportional to its score with respect to the current query.

Within such *sampled displays* both documents with high and low ranking have a non-zero probability of being included, thus exhibiting more variability and enabling the user to direct the search away from a local maximum. We expect that a sampled display strategy will be useful in situations where the initial query is imprecise, i.e., when the target document is ranked very low in the search result list.

The situation of using small display sizes for search makes the problem similar to the task of Adaptive Information Filtering where the importance of the interplay between *exploitation* and *exploration* has been recognized. It is to be expected that other more optimal sampling strategies exist which provide a better balance between exploitation and exploration. Providing these *preliminary* results we illustrate both the need and effectiveness of such strategies.

4 Experimental Procedure

In order to quantify the effect of relevance feedback and alternative display strategies, we need to define (i) the search task, (ii) the evaluation methodology and (iii) the initial conditions. These issues are discussed in Sections 4.1-4.3.

In the experiments we use the Reuters-21578 collection of textual documents. From the documents we extract the contents of the two fields, the "Body" and the "Title" and after removing the stop words we create vector representation of documents with *tf-idf* weights. Since some of the documents in the collection have empty "Body" fields, we removed them from the collection and arrived at a data set of 19,043 documents.

4.1 Task Model

In the context of retrieval, at least three classes of search may be identified [5]:

- Target document search – the user's information need is satisfied by a particular document. For example, a researcher may be looking for a particular paper on a research topic.
- Category search - the user seeks one or more items from a general category or a topic. This task places more emphasis on the semantic content of the data and often requires subjective judgements.
- Open ended browsing – the user has some vague idea of what to look for but is open to exploration and may repeatedly change topic during search.

Of these three scenarios, the target document search (or known-item search) is most amenable to evaluation for there are several clear measures of effectiveness including the total time or the total number of documents examined before the target is found.

We chose to compare different systems based on the total number of documents examined before the target is found. For comparison purposes, this number is compared with the rank of the document after the initial query, i.e. before any relevance feedback is applied. This rank is the number of documents that a user must examine when scrolling and no feedback is provided.

In the context of target search, we restrict a user's actions to selecting one of the K documents currently being displayed. Thus, there are K possible user actions in each iteration.

While target document search is typically equated with the 'known item search', the former encompasses a wider spectrum of search scenarios. It can include any information search that is satisfied with a specific document, regardless of whether or not the user is familiar with the target document. So long as the user can recognize that his or her information need is satisfied when the specific document is displayed, we can model this as target document search.

4.2 Evaluation Methodology

The experimental procedure to examine the effect of relevance feedback and alternative display strategies is designed to include the complete space of possible user's interactions with the system within the particular scenario. This is possible because of the small number of documents K that are displayed at each iteration. Thus, we can examine all user's strategies, including the optimal performance of an 'ideal user' whose selections minimize the number of documents that must be examined before identifying the target.

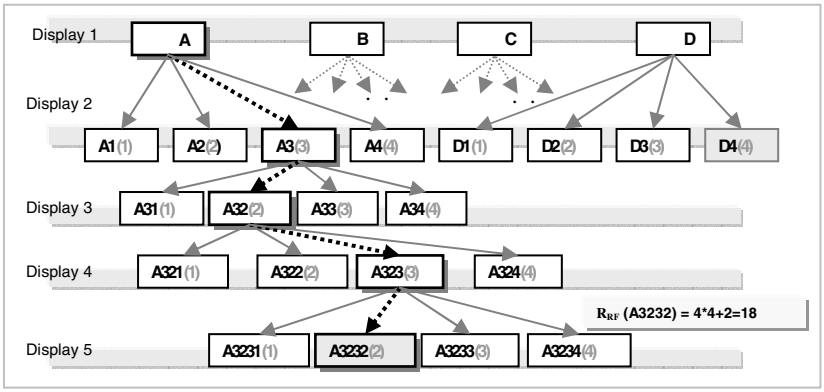

Fig. 2. Decision tree for iterative relevance feedback, showing nodes in which the target document is reached, the rank of a document within each display, and the calculation of RF-rank for the target. Expansion of this branch has stopped at depth five because the target has been found

At each iteration, the tree expands by a factor of K (See Figure 2). For practical purposes, we limit the depth of the tree to depth five, i.e., an initial display of K documents followed by five iterations of relevance feedback. For K=4, the maximum number of nodes in the tree is $1+4+4^2+4^3+4^4+4^5=1365$, where a node represents a display of K documents. The tree may be smaller if the target is located earlier since branches of the tree are not expanded once the target has been displayed. The choice of using a display size of four is made keeping in mind the display size of a typical mobile device. To account for a variety and range of such devices, a range of display size could be investigated using the same methodology

The *minimum rank* for a given target document corresponds to the best case scenario where the user always provides the system with the optimal document for relevance feedback. It is important to note that 'optimal' may not always mean the document most similar to the target.

We can also examine the *number of target document occurrences* in a tree. This provides a measure of the likelihood of a non-ideal user locating the target document. For example, if the target document appears in only one path of the tree, then any deviation by a real user from the "ideal" would result in a failed search. Conversely, if the target document appears in many paths, then deviations from the "ideal" are still likely to yield successful searches, albeit that these searches require further effort. Examining the set of documents displayed after each iteration can also reveal properties of the relevance feedback and/or display strategy.

Finally, since the trees are generated automatically with no user interaction, it is possible to generate a very large number of trees, thereby facilitating statistical analysis.

4.2.1 Construction of the User Decision Trees

Figure 2 illustrates a tree that represents the space of all user decisions. At each iteration, the tree expands by a factor of four. While the general behaviour of relevance feedback algorithms is of interest, understanding the impact of relevance feedback to the first few iterations is most important from the point of view of real applications – we therefore limit the expansion to depth five of the tree (the root is at depth zero).

The initial display of four documents is labelled A-B-C-D and is followed by five iterations of relevance feedback. At each iteration, selection of a document from the display leads to a new branch in the tree. Some branches contain the target document. Since we are focussing on the target document search, the branches below the level for which at least one node is the target document need not be expanded further (see Figure 2; the second level of the sub-tree starting with D, which contains the target document at node D4).

We annotate each document in the graph by its rank *r* within the display of K=4 documents, with r having the value r = 1, 2, 3, or 4. We concatenate displays from relevance feedback iterations by appending to the list the most recent display. The resulting list shows documents in the order in which the user would view them. For each document in the graph, we can identify the corresponding ranked list and calculate the *relevance feedback rank* $R_{RF}=d\cdot K+r$, where d is the number of previous dis-

plays, d = 0, 1, 2, 3, 4 or 5. R_{RF} essentially corresponds to the number of documents that the user has viewed before locating the document. In our evaluations we compare R_{RF} of the document with its rank in the baseline ranked list obtained from the initial query. We refer to this baseline rank as the *scroll rank*, R_{Scroll}, since this is the number of documents that the user would have to examine by scrolling down the original search result in order to reach the target document.

The task is therefore similar to the Ostensive Retrieval Model [3], except that we use standard relevance feedback algorithms between two displays. Very recently, [26] dealt with the question of measuring the performance of implicit feedback models by conducting a simulation-based evaluation.

4.3 Initialisation

We begin experiments by randomly selecting a target document from the database. An initial query is then automatically generated by randomly selecting M terms from the target document. In our experiments M=4. These M terms are used in two ways: as a search query to obtain the baseline search results and as input to the relevance feedback procedure which will further refine the query based on the user's responses. Using four query terms is higher than the average even in internet search engines - the development of predictive texting features in these devices makes this number reasonable. The query vector is simply a vector of equally weighted terms, reflecting our assumption that the user may have some expectations of finding certain terms in the document but is otherwise unaware of the characteristics of the target document or the document corpus in general. The user's relevance feedback iterations start with an initial display of K documents that are chosen based on which display strategy is being used.

The user's response is used by the relevance feedback algorithm to modify the query. The documents in the collection are then scored against the new query and a new display of K documents is presented to the user, based on the search ranking and display strategy. Previously viewed documents are not included in the subsequent search iterations.

5 Results

In our experiments we generated 100 trees, corresponding to 100 distinct target documents, randomly selected from the subset of 19,043 documents from the Reuters collection. The initial query was composed of four random terms present in the target document and the scroll rank of each target document was recorded.

For each target document we generated a complete search tree based on iterative feedback, with two types of displays: (1) the Top-K display always showing the top 4 ranked documents from the search iteration and (2) the Sampled display that probabilistically selects the documents based on the current ranking of documents in the database. Trees and paths within the trees that contain the target documents are referred to as *successful searches* for the relevance feedback scheme. Tables 1-4 summarize the statistics of the tree displays and successful searches.

Table 1. Statistics about search tree results for three feedback algorithms and the two display strategies

	Rocchio Feedback Algorithm		RSJ Feedback Algorithm		Bayesian Feedback Algorithm	
	Top-K Display Scheme	Sampled Display Scheme	Top-K Display Scheme	Sampled Display Scheme	Top-K Display Scheme	Sampled Display Scheme
Percentage of trees with target	52	97	39	33	52	90
Percentage of paths containing the target	46.67	4.5	27.99	0.087	46.80	4.30
Average R_{Scroll} of targets found in trees	13.79	98.54	37.28	312.03	7.92	64.23
Average min R_{RF} of targets found in trees	6.5	11.25	7.20	17.76	6.13	10.61
Average R_{RF} for the'average user'	20.53	20.2	20.22	18.26	21.27	19.94

Table 2. Performance of the Rocchio RF Algorithm based on the Initial Query

Scroll Rank Range	Number of Targets	Number of Targets Found		Avg. No. of Documents viewed without RF		Avg. No. of Documents viewed by the 'ideal user' using RF		No. of Documents viewed with RF averaged over all successful users	
		Top-K	Sampled	Top-K	Sampled	Top-K	Sampled	Top-K	Sampled
1 – 20	45	45(100%)	45(100%)	4.38	4.38	4.31	5.33	16.54	19.13
21 – 40	14	6(42.8%)	14(100%)	25.5	29.79	20.67	13.07	21.62	21.92
41 – 60	5	0(0%)	5(100%)	-	54.2	-	16.6	-	21.99
61 – 80	4	0(0%)	4(100%)	-	66.5	-	16.5	-	21.80
81 – 100	6	0(0%)	6(100%)	-	92.83	-	15.33	-	21.49
101 – Last Rank	26	1(3.84%)	23(89%)	367	341.3	20	18.56	20.78	22.14

Table 3. Performance of the RSJ RF Algorithm based on the Initial Query

Scroll Rank Range	Number of Targets	Number of Targets Found		Avg. No. of Documentsviewed without RF		Avg. No. of Documents viewed by the 'ideal user' using RF		No. of Documents viewed with RF averaged over all successful users	
		Top-K	Sampled	Top-K	Sampled	Top-K	Sampled	Top-K	Sampled
1 – 20	27	27(100%)	7(25.93%)	5.67	4.72	4.26	17	19.21	18.67
21 – 40	6	2(33.33%)	2(33.33%)	34	31	7.5	17	12.46	17
41 – 60	5	3(60%)	3(60%)	47.33	41.67	6.33	17.33	7.4	17.33
61 – 80	8	1(12.5%)	3(37.5%)	74	68.33	17	21	18.15	21
81 – 100	2	1(50%)	2(100%)	81	88	24	17	24	17
101 – Last Rank	52	5(9.61%)	16(30.77%)	187.2	606	18.2	17.5	21.72	17.94

Table 4. Performance of the Bayesian RF Algorithm based on the Initial Query

Scroll Rank Range	Number of Targets	Number of Targets Found		Avg. No. of Documents viewed without RF		Avg. No. of Documents viewed by the 'ideal user' using RF		No. of Documents viewed with RF averaged over all successful users	
		Top-K	Sampled	Top-K	Sampled	Top-K	Sampled	Top-K	Sampled
1 – 20	45	45(100%	45(100%)	4.38	4.38	4.31	5.02	16.54	18.75
21 – 40	14	6(42.8%	14(100%)	25.17	29.78	17.67	13.07	22.21	21.35
41 – 60	5	0(0%)	5(100%)	-	54.2	-	13.4	-	21.52
61 – 80	4	1(25%)	4(100%)	64	66.5	17	18.5	18.05	21.98
81 – 100	6	0(0%)	6(100%)	-	92.83	-	18.33	-	22.18
101 – Last Rank	26	0(0%)	16(61.53%)	-	254.56	-	18.44	-	21.92

6 Discussion

6.1 Top-K Display Scheme

The number of documents *seen* without relevance feedback(RF) is the scroll rank of the target in the initial ranked list. The RF rank of an *ideal user* is the minimum path length from the root of the tree to a node with the target, whereas the mean length of all paths leading to the target represents the average performance of a successful user. The first row in Table 1 is the probability that a search (using a given display scheme) will be successful, and row two is the probability that a non-ideal user will find the target. For the Top-K display strategy, about 50% of the trees contain the target (lower for RSJ). In the remaining cases, the target was not found within five rounds of relevance feedback. This percentage is clearly a function of the accuracy of the initial query, which can be judged by examining the scroll rank of the target document.

For the Rocchio and Bayesian algorithms, we see that for a scroll rank of less than 20, relevance feedback with Top-K display is successful 100% of the time. For higher values of the initial scroll ranks, i.e.; poor queries, we observe a fall off in the percentage of successful searches. However, the sampled display scheme offers performance that is more or less constant. For the case of RSJ, which explicitly incorporates a term expansion strategy, the Top-K display strategy performed better.

The ideal user represents the best possible performance achievable. Real users are unlikely to perform as well. However, the average number of paths in the tree that contain the target suggests that deviations from the ideal still have a reasonable chance of locating the target document. The average rank of target documents in the tree was obtained by calculating first the average rank for the target document within its particular tree and then averaged over the set of all the trees that contain target documents.

6.1.1 Convergence

It was observed that sub-trees below a node at depth 4 were identical. That is, the set of four documents displayed to the user at depth 5 was the same, irrespective of the choice of relevant document at the preceding level. Note that the relative *order* of

displayed four documents may be affected by the relevance feedback, but the *same* documents appeared in all four sub-trees. It is important to note that the convergence was observed for all three algorithms: even though the sets to which they converged were different.

Since the phenomenon was not symptomatic of any one particular algorithm, we suspect that this *convergence* is due to the greedy nature of the display updating strategy – that of picking the K most probable items (based on the score with respect to the current query). Since the aim of the RF algorithm is to extract *similar* documents from the collection, it results in a situation where successive displays offer no diversity. The small variation across the documents in the display is also due to the small number of documents, 4, in the display. However, similar convergence properties were observed for larger displays.

6.2 Sampled Display Scheme

For the alternative display, a higher percentage of the trees contained the target document with the *conservative* Rocchio and Bayesian schemes. More importantly, we do not observe a performance degradation as the quality of the initially query degrades. And for very poor initial queries, the alternative display strategy is superior. Since the RSJ algorithm itself considers exploring different regions of the search space by query expansion, use of the Sampled display strategy led to an over-adventurous approach, resulting in a smaller number of successful searches and fewer paths leading to the target in a given tree. This illustrates the classical dilemma between exploration and exploitation.

Analysis of the trees containing the target revealed that the average scroll rank was much higher than the rank for an ideal user using relevance feedback and the alternative display, representing a very significant reduction in the number of documents examined. However, once more, we need to recognize that real users are unlikely to perform as well as the ideal user. For the sampled display, the average number of paths in the tree that contain the target is low, which suggests that deviations from the ideal may have a significant detrimental effect on performance.

Finally, we note that the convergence phenomenon observed with the Top-K display was not exhibited using the Sampled display.

6.3 Comparing the Feedback Algorithms

When comparing the 3 feedback algorithms, we find that the following points stand out:

1) The performance of the Rocchio algorithm and the Bayesian one are very similar. We hypothesize that this is mainly because of the fact that in this implementation, they both use the Cosine similarity metric.
2) Even with the same queries (the same terms chosen from the same targets), the RSJ algorithm produces a very different initial ranking because it uses the BM25 ranking algorithm.
3) RSJ uses a specifically constructed term expansion strategy, which results in the feedback process itself working – shown by the fact that even with the cases where the initial scroll rank is low and the Top-K display update is used, RSJ still man-

ages to find the target in a few cases. The sensitivity to feedback in this case is reflected in the smaller number of paths with the target, as compared to similar runs for Rocchio and the Bayesian algorithm.

4) The default values of parameters were used in all three algorithms. While the 'K' and 'b' values for RSJ are more or less generally accepted values for similar situations, the α and β values in Rocchio (which control the relative effect of past and present feedback provided) and the σ in the Bayesian algorithm (which loosely controls the noise associated with the current feedback) can be tuned to alter the results.

7 Conclusions

We examined whether relevance feedback and alternative display strategies can be used to reduce the number of documents that a user of a mobile device with limited display capabilities has to examine before locating a target document. In this scenario, it is possible to construct a tree representing all possible user actions for a small number of feedback iterations. This allows us to determine the performance of an "ideal" user, i.e. no real user can perform better. We are therefore able to establish an upper limit on the performance improvement such systems can deliver. To the best of our knowledge, this has not previously been done. The experimental paradigm has the further advantages of (i) not requiring a real user study, which can be time consuming, and (ii) the ability to simulate very many searches, thereby facilitating statistical analysis.

Using each of three relevance feedback algorithms with a display size of four documents, we constructed 100 trees. With a greedy display strategy, analysis of the trees containing the target(i.e; the successful searches) revealed that relevance feedback with a greedy display strategy resulted in close to 50% reduction in the number of documents that a user needed to examine compared with simply performing a linear search of a ranked list calculated from the initial query. It should however be noted that this number is exaggerated because of the presence of outliers.

It is unclear as to why the improvement is so low. This may be due to the experimental procedure which required a user to always select one document as relevant, even if none of the displayed documents actually was relevant. Future work is needed to examine whether performance can be improved by:

1. alternative values for the algorithm parameters
2. the identification of non-relevant as well as relevant documents
3. alternative distance metrics

Similarly, the formation of the initial query by selection of random terms from the target document should also be examined. Experiments in which the query is created by selecting terms which occur most or more frequently are obvious directions for investigation. The observation of convergence of the relevance feedback algorithm using a greedy display also needs investigation. More positively, it was observed that relevance feedback almost never led to worse performance for an ideal user.

We also examined how the performance of the system was affected by an alternative display strategy in which the displayed documents were drawn with the same underlying distribution as the current scores of documents in the database. This sam-

pling strategy crudely approximates a strategy in which we attempt to maximize the immediate information gain.

Using this display strategy, the Rocchio algorithm (with no explicit feature selection) and the Bayesian algorithm (which implicitly uses all the features incorporated into the distance metric) had a larger number of successful searches. However, this large improvement may be misleading. Firstly, the target is present in an extremely small fraction of the 1024 paths of the tree. Thus, while the "ideal" user is guaranteed to find the target, any deviation by real users from the "ideal" is likely to result in a failed search. RSJ's offer weight selection mechanism is known to be unstable, and coupling this with an exploratory display update strategy led to worse performance.

Generalizing, it is clear that if the user's query is sufficiently accurate, then the initial rank of the target document is likely to be high and scrolling or relevance feedback with a greedy display performs almost equally well. However, if the user's initial query is poor, then scrolling is futile and relevance feedback is required – either with a display strategy that explores larger regions of the search space or a feedback algorithm that does the same.

The end result of our investigations is that inclusion of Relevance Feedback into the retrieval process is not, on average, likely to drastically improve the retrieval effectiveness. It would however by interesting to measure how the utilization of more complicated inter-document properties (apart from the simple cosine distance metric) affects the performance gain. Other future work includes the examination of other display strategies, including hybrid strategies that attempt to optimally combine the exploratory properties of maximizing information gain with the exploitative properties of greedy displays.

References

[1] Buyukkokten, O., Garcia-Molina, H., Paepcke, A. and T. Winograd. Power Browser: Efficient Web Browsing for PDAs. In the Proceedings of the ACM Conference on Computers and Human Interaction (CHI'00), 2000.

[2] Buyukkokten, O., Garcia-Molina, H., and Paepcke, A. Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices. In the Proceedings of the Tenths International World Wide Web Conference (WWW 10), 2001.

[3] Campbell, I. & van Rijsbergen, C.J. (1996). The Ostensive model of developing information needs. In: Ingwersen, P. & Pors, N.O. (eds.). Information Science: Integration in Perspective. Proceedings of CoLIS 2, p. 251-268

[4] Chen, Y., Ma, W-Y., and Zhang, H-J. Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices. In the Proceedings of the Twelfth World Wide Web Conference, Budapest, May 2003 (to appear).

[5] Cox, I. J., Miller, M.L., Minka, T.P., Papathomas, T.V., and Yianilos, P.N. The Bayesian Image Retrieval System, PicHunter: Theory, Implementation and Psychophysical Experiments. IEEE Transactions on Image Processing, 9(1):20-37, 2000.

[6] Evans, D.A. and Lefferts, R.G. CLARIT-TREC experiments. Information Processing and Management, 31(3):494-501, 1995.

[7] Grossman, D.A., Frieder, O., Holmes, D.O., and Roberts, D.C. Integrating Structured Data and Text: A Relational Approach. Journal of the American Society of Information Science, 48(2), February 1997.

[8] Harman, D. Relevance feedback and other query modification techniques. In W. Frakes and R. Baeza-Yates, editors. Information Retrieval. Data Structures and Algorithms. Pages 131-160. Prentice Hall, 1992.

[9] Harman, D. Relevance feedback revisited. Proceedings of 15th annual international ACM SIGIR conference on research and development in information retrieval, Copenhagen, 1.10, 1992.

[10] Jones, M., Marsden, G., Mohd-Nasir, N., and Boone, K., and Buchanan, G. Improving Web Interaction on Small Displays. In the Proceedings of the 8th World Wide Web Conference, Toronto, Canada, May 1999.

[11] Jones, M. and Marsden, G., From the Large Screen to the Small Screen. Retaining the Designer's Design for Effective user Interaction. In IEEE Colloquium on Issues for Networked Interpersonal Communicators. 239(3), pp 1-4., 1997.

[12] Lewis and Ringuette, 1994 David Lewis and Marc Ringuette. A comparison of two learning algorithms for text categorization. In Symposium on Document Analysis and Information Retrieval, 1994.

[13] Magennis, Mark, and van Rijsbergen, Cornelis J. The potential and actual effectiveness of interactive query expansion. Proceedings of 20th annual international ACM SIGIR conference on research and development in information retrieval, Philadelphia, 1997.

[14] Milic-Frayling, N. and R. Sommerer, R., SmartView: Enhanced document viewer for mobile devices. Microsoft Technical Report MSR-TR-2002-114, November 2002.

[15] Over, P. TREC-5 interactive track report. In Proceedings of the Fifth Text REtrieval Conference (TREC-5), 1996.

[16] Overview of the Fifth Text REtrieval Conference (TREC-5). Edited by D.K. Harman. Gaithersburg, MD: NIST, 1997.

[17] Rocchio, J. Relevance feedback information retrieval. In Gerard Salton (ed.): The Smart Retrieval System — Experiments in Automatic Document Processing, pp. 313–323. Prentice-Hall, Englewood Cliffs, N.J., 1971.

[18] Robertson, S.E., Sparck Jones, K. Relevance weighting of search terms. Journal of the American Society for Information Science 27, 1976, pp. 129-146.

[19] Robertson, S.E. at al. Okapi at TREC-3. In Overview of the Third Text Retrieval Conference (TREC-3). Edited by D.K. Harman. Gaithersburg, MD: NIST, 1995 (NIST Special Publication 500-225).

[20] Robertson, Stephen and Hull, David A. The TREC-9 Filtering Track Final Report. In NIST Special Publication 500-249: The Ninth Text Retrieval Conference (TREC-9). Edited by E.M. Voorhees and D.K. Harman, Gaithersburg, MD, 2000.

[21] Rodden, K., Milic-Frayling, N., Sommerer, R., & Blackwell, A., Effective Web Searching on Mobile Devices. In the *Proceedings of the HCI Conference*, Bath, September 2003.

[22] Ruvini, J-D. Adapting to the User's Internet Search Strategy. IUI'03, Miami, Florida, January 12-15, 2003.

[23] Sellen, A.J., Murphy, R., and Shaw, K.L., How knowledge workers use the web, in D. Wixon (ed.), Proceedings of CHI 2002, ACM, pp 227-234, 2002.

[24] Sparck Jones, K., Walker, S., and Robertson, S.E. A probabilistic model of information retrieval: development and comparative experiments. Information Processing and Management 36 (2000) 779-808, 809-840.

[25] Vinay, V, Cox, I J, Milic-Frayling, N, Wood, K. Evaluating Relevance Feedback and Display Strategies for Searching on Small Displays. SPIRE 2004

[26] White, R.W, Jose, J.M, van Rijsbergen, Cornelis J and Ruthven I, A Simulated Study of Implicit Feedback Models, ECIR 2004