A Simulated Study of Implicit Feedback Models

Ryen W. White[1], Joemon M. Jose[1], C.J. van Rijsbergen[1] and Ian Ruthven[2]

[1] Department of Computing Science,
University of Glasgow. Glasgow, Scotland. G12 8RZ.
{ryen,jj,keith}@dcs.gla.ac.uk

[2] Department of Computer and Information Sciences,
University of Strathclyde. Glasgow, Scotland. G1 1XH.
ir@cis.strath.ac.uk

Abstract. In this paper we report on a study of implicit feedback models for unobtrusively tracking the information needs of searchers. Such models use relevance information gathered from searcher interaction and can be a potential substitute for explicit relevance feedback. We introduce a variety of implicit feedback models designed to enhance an Information Retrieval (IR) system's representation of searchers' information needs. To benchmark their performance we use a simulation-centric evaluation methodology that measures how well each model learns relevance and improves search effectiveness. The results show that a heuristic-based binary voting model and one based on Jeffrey's rule of conditioning [5] outperform the other models under investigation.

1 Introduction

Relevance feedback (RF) [11] is the main post-query method for automatically improving a system's representation of a searcher's information need. The technique relies on explicit relevance assessments (i.e. indications of which documents contain relevant information), and creates a revised query attuned to those documents marked. The need to explicitly mark relevant documents means searchers may be unwilling to directly provide relevance information.

Implicit RF, in which an IR system unobtrusively monitors search behaviour, removes the need for the searcher to explicitly indicate which documents are relevant. The technique uses implicit relevance indications, gathered from searcher interaction, to modify the initial query. Whilst not being as accurate as explicit feedback, in previous work [14] we have shown that implicit feedback can be an effective substitute for explicit feedback in interactive information seeking environments. In this paper we evaluate the search effectiveness of a variety of implicit models using a simulation-based methodology. This strategy, similar to [6,9], is not affected by inter-searcher inconsistencies, is less time consuming and costly, and allows environmental and situational variables to be more strictly controlled. It allows us to compare and fine-tune the various models before they are employed in a real system. We use simulations since no precedent has yet been set on how to best evaluate implicit feedback models.

We investigate a variety of different methods of relevance feedback weighting based on implicit evidence. The implicit feedback models presented use different methods of handling this implicit evidence and updating their understanding of searcher needs in light of it. The study compares the models' ability to learn relevance and create more effective search queries.

The remainder this paper is structured as follows. In Section 2 we describe the document representations and relevance paths used to create evidence for the models described in Section 3. In Section 4 we describe the simulations used to test our approach, the results in Section 5, and conclude in Section 6.

2 Document Representations and Relevance Paths

The implicit models we evaluate in this paper gather relevance information from searchers' exploration of the *information space*; the information content of the top-ranked retrieved document set. This space is created at retrieval time and is characterised by the presence of search terms (i.e. it is query-relevant). Exploring it allows searchers to deeply examine search results and facilitates access to potentially useful information. Searchers can interact with *document representations* and follow *relevance paths* between these representations, generating evidence for the implicit models we evaluate. A similar granular approach has been shown to be effective in previous studies [16].

2.1 Document Representations

Documents are represented in the information space by their full-text and a variety of smaller, query-relevant representations, created at retrieval time. These include the document title and a four-sentence query-biased summary of the document [15]; a list of *top-ranking sentences* (TRS) extracted from the top thirty documents retrieved, scored in relation to the query, and; each summary sentence in the context it occurs in the document (i.e. with the preceding and following sentence). Each summary sentence and top-ranking sentence is regarded as a representation of the document. Since the full-text of documents can contain irrelevant information, shifting the focus of interaction to the query-relevant parts reduces the likelihood that erroneous terms will be selected by the implicit feedback models.

2.2 Relevance Paths

The six types of document representations described in Section 2.1 combine to form a *relevance path*. The further along a path a searcher travels the more relevant the information in the path is assumed to be. The paths can vary in length from one to six representations, and searchers can access the full-text of the document from any step in the path. *Relevance paths can start from top-ranking sentences or document titles*. Certain aspects of the path order are fixed e.g. the searcher must view a summary sentence before visiting that sentence in context. Figure 1 illustrates an example relevance path on an experimental search interface based on [16].

Some representations of each document are fixed in content, i.e. the title and full-text of the document, whereas other representations, such as the summary, are

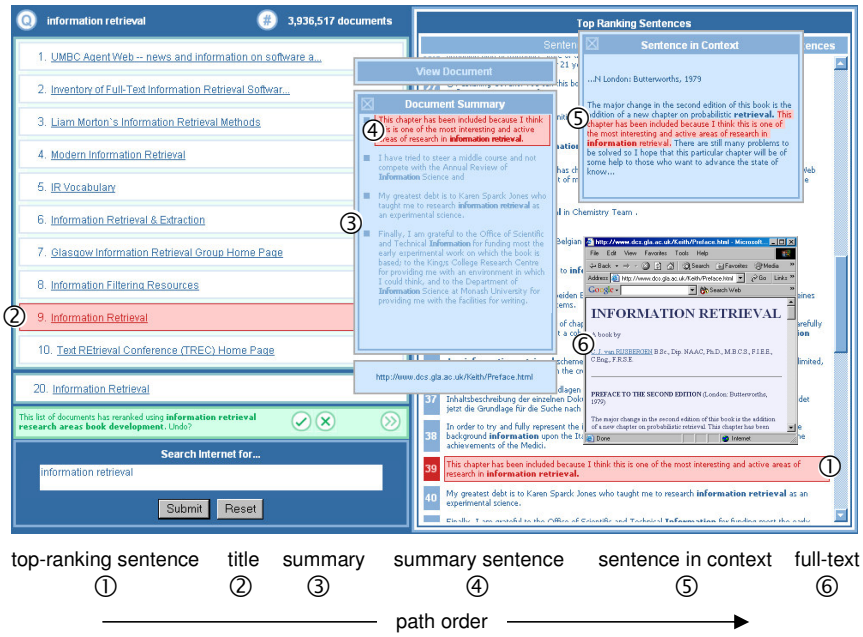


Fig 1. The relevance path

dependent on the query and hence variable in content. Therefore, for each document, there may be many *potential* relevance paths. We use the distance travelled along the path and the particular representations viewed as evidence for the implicit models described in the next section.

3 Implicit Feedback Models

We developed six different implicit models that will be discussed in this section. The relevance assessments in *all* models are obtained implicitly, by interpreting a searcher's selection of one information object over others as an indication that this object is more relevant.

We encourage searchers to deeply examine the results of their search, following relevance paths and *exploring* the information space. All approaches use this exploration as a source of implicit evidence and choose the potentially relevant terms to expand the query. The presence of an information space allows certain models to retain some memory of searcher preferences and behaviour. This memory facilitates *learning* (i.e. the models learn over time what terms are relevant). The models presented learn in different ways, and in this section we describe each of them. All models, with the exception of the document-centric approach described in Section 3.3.1, use the document representations and relevance paths described in Section 2.

3.1 Binary Voting Model

The *binary voting model* [16] is a heuristic-based implicit feedback model. To identify potentially useful expansion terms the model allows each document representation to 'vote' for the terms it contains. When a term is present in a viewed representation it receives a 'vote', when it is not present it receives no vote. All terms are candidates in the voting process, and these votes accumulate across all viewed representations.

Different *types* of representation vary in length and may have a different indicative worth, typically measured based on representation length [1]. For example, a top-ranking sentence is shorter than a query-biased document summary (typically composed of four sentences) and is therefore less indicative of document content. To compensate for this, we use heuristic weights for the indicative worth of each type of representation. The weights used are 0.1 for title, 0.2 for top-ranking sentence, 0.3 for summary, 0.2 for summary sentence and 0.2 for sentence in context. These weights, based only on the *typical* length of a representation, ensure that the total score for a term in a relevance path is between 0 and 1 (inclusive).

The terms with the highest overall vote are those that are taken to best describe the information viewed by the searcher (i.e. those terms that are present most often across all representations) and can be used to approximate searcher interests.

3.2 Jeffrey's Conditioning Model

The next implicit model discussed uses *Jeffrey's rule of conditioning* [5] to revise the probability of term relevance in light of evidence gathered from searcher interaction. Jeffrey's conditioning captures the uncertain nature of implicit evidence, and is used since even after the passage of experience (i.e. following a relevance path) the model is still uncertain about the relevance of a term. The approach we use for this revision is based on that proposed by van Rijsbergen [12].

The binary voting model used a set of pre-defined heuristic weights for the indicativity of a path's constituent representations. In the Jeffrey's conditioning model we use various measures to describe the value, or worth, of the evidence a representation provides. We combine a confidence measure that uses the relative position of representations in the relevance path with a measure of indicativity based on the concepts in a representation. In this section we describe each of these measures, and how the Jeffrey's conditioning model weights potential expansion terms.

3.2.1 Path Weighting

For each path, we become more confident about the value of aged relevance information as we regress. In our approach we assign an exponentially increasing relevance profile to *aged* relevance. The representations that comprise the path are smaller than documents, the paths are generally short (i.e. no more than six representations) and the most recent document representation is not necessarily the most relevant.

The assumption we make is that the further we travel along a relevance path, the more certain we are about the relevance of the information towards the start of the path. As the viewing of the *next* representation is exploratory and *driven by curiosity*

as well as information need we are cautious, and hence less confident about the value of this evidence. This confidence, c, is assigned *from the start of the path* to each representation i,

$$c_i = \frac{1}{2^i}, \text{ where } i \geq 1 \tag{1}$$

However, since across a whole path, the values of c_i do not sum to one, we must normalise and compute the confidence c for each representation i in a path of length N using,

$$c_i = \left(\frac{1}{2^i} + \frac{1}{N.2^N} \right), \text{ where } \sum_{i=1}^{N} c_i = 1 \text{ and } i \in \{1, 2, ..., N\} \tag{2}$$

3.2.2 Indicativity and Quality of Evidence

In the previous section we described the confidence in the relevance of representations based on their position in the relevance path. The quality of evidence in a representation, or its *indicative worth*, can also affect how confident we are in the value of its content. In the binary voting model we use heuristics based on the *typical* length of document representations to measure indicativity. However, titles and top-ranking sentences, which may be very indicative of document content, are short and will have low indicativity scores if their typical length is the attribute used to score them.

In this approach, we use the non-stopword terms, or *concepts*, in a representation instead of representation length. We weight a term t in document d using its normalised term frequency [4], and the sum of all weights in a document is 1. The larger this value, the more often it occurs in the document, and the more representative of document content that term can be seen to be. To compute the indicativity index I for a representation r we sum the weight of a term in a document $w_{t,d}$ for all *unique* terms in r,

$$I_r = \sum_{t \in r} w_{t,d} \tag{3}$$

The I_r ranges between 0 and 1, is never 0, and is 1 only if the representation contains every unique term in the document. The indicativity measure is only incremented if there is a match between the unique terms in the document and those in the representation[1].

Relevance paths will contain representations of varying quality. We compute the *value* of the evidence in a representation by multiplying its indicativity by its confidence. Using these measures ensures that the worthwhile representations in each relevance path contribute most to the selection of potentially useful query expansion terms. In the next section we describe how such terms are chosen.

---

[1] This measure is similar to a *Hamming distance* [3], but uses term *weights*, rather than presence/absence.

3.2.3 Term Weighting

The Jeffrey's model assumes the existence of a *term space T*, a mutually exclusive set of all (non-stemmed, non-stopword) terms in the information space. Each term in *T* is independent and has an associated frequency in the information space. We define the probability that a term *t* is relevant based on a probability distribution *P* over *T* as,

$$P(t) = \frac{ntf(t)}{\sum_{t \in T} ntf(t)} \quad \begin{array}{l}\text{where } ntf(t) \text{ is the } \textit{normalised term frequency} \text{ [4]} \\ \text{of term } t \text{ in the term space } T\end{array} \tag{4}$$

To update this probability based on new evidence gathered from interaction we use *Jeffrey's Rule of Conditioning*, applied at the end of each relevance path. We consider this relevance path *p* as a new source of evidence to update the probability to say *P'*.

The viewing of a representation p_i creates new evidence for the terms in that representation. We use Jeffrey's rule of conditioning to update the probabilities based on this new evidence using the following formula,

$$P'(t) = \left[P(t=1 \mid p_i) \frac{P'(t=1)}{P(t=1)} + P(t=0 \mid p_i) \frac{P'(t=0)}{P(t=0)} \right] . P(t) \tag{5}$$

This estimation calculates the revised probability of relevance for a term *t* given a representation p_i, where $P(t = 1)$ is the probability of observing *t*, and $P(t = 0)$ the probability of not observing *t*. This updated probability reflects the 'passage of experience' and is similar to that described in [12].

A relevance path contains a number of representations. We update the probabilities after the traversal of a relevance path. The length of a relevance path ranges between 1 and 6 steps. We denote this length using *N*. When this length is greater than one we update the probabilities across this path. The probability of relevance of a term across a path of length *N* is denoted P_N and given through *successive updating*,

$$P_N(t) = \sum_{i=1}^{N-1} c_i . I_i . \left[\left(P_i(t=1 \mid p_i) \frac{P_{i+1}(t=1)}{P_i(t=1)} + P_i(t=0 \mid p_i) \frac{P_{i+1}(t=0)}{P_i(t=0)} \right) . P_i(t) \right] \tag{6}$$

where a representation at step *i* in the path *p* is denoted p_i . The confidence in the value of the representation is denoted c_i and I_i is the indicativity of the representation. In this equation, the order of the updating matters, so the order in which the searcher traverses the path also matters.

The actual revision of the probabilities will occur after each path. Once learned, the probabilities of relevance remain stable until the next revision (i.e. the next relevance path). Only terms in *T* that appear in the relevance path will have their probabilities revised *directly* [2].

[2] Based on the new evidence probabilities are redistributed to make the sum 1.

3.3 WPQ-Based Models

In this section we present three implicit feedback models that use the popular *wpq* method [8] to rank terms for query expansion. This method has been shown to be effective and produce good results. The equation for *wpq* is shown below, where the typical values r_t = the number of seen relevant documents containing term t, n_t = the number of documents containing t, R = the number of seen relevant documents for query q, N = the number of documents in the collection.

$$wpq_t = \log \frac{r_t/(R-r_t)}{(n_t-r_t)/(N-n_t-R+r_t)} \cdot \left(\frac{r_t}{R} - \frac{n_t-r_t}{N-R} \right) \quad (7)$$

The *wpq* method is based on probabilistic distributions of a term in relevant and non-relevant documents. As the values of r_t and R change during searcher interaction, the *wpq*-generated term weights also change. However, there is no retained memory of these term weights between iterations, and wpq_t is recomputed after each iteration. The *wpq* approaches learn what *information objects* are relevant but do not directly 'remember' the weights assigned to *terms*. This is unlike the Jeffrey's and binary voting models, which store and revise term weights for the entire search session.

3.3.1 WPQ Document Model

The *wpq document model* uses the full-text of documents, rather than granular representations or paths that link them. The *wpq* formula is applied to each document and expansion terms chosen from it. The values of R = the number of seen documents, r_t = the number of seen documents containing term t, N = the number of top-ranked documents and n_t = the number of top-ranked documents containing the term t. This approach is effectively a traditional explicit relevance feedback model, choosing one relevant document per iteration. This is a realistic model since implicit feedback is typically gathered sequentially (i.e. one relevance indication after another) and was included in the study to investigate the effects of using whole documents for such feedback.

3.3.2 WPQ Path Model

In the *wpq path model* the terms from each complete relevance path are pooled together and ranked based on their *wpq* score. We use the values R = the number of seen paths, r_t = the number of seen paths containing term t, N = the total number of paths generated from the top 30 retrieved documents, n_t = the number of generated paths that contain the term t. Since it uses terms in the *complete path* for query expansion, this model does not use any path weighting or indicativity measures. This model was chosen to investigate combining *wpq* and relevance paths for implicit feedback.

3.3.3 WPQ Ostensive Profile Model

The *wpq ostensive profile model* considers each representation in the relevance path separately, applying the *wpq* formula and ranking the terms each representation contains. This model adds a temporal dimension to relevance, assigning a within-path *ostensive relevance profile* [2] that suggests a recently viewed step in the relevance path is more indicative of the current information need than a previously viewed one. This differs from the Jeffrey's model, which assigns a reduced weight to most recently

viewed step in the path. The *wpq* weights are normalised using such a profile. The model treats a relevance path a series of representations, and uses each representation separately for *wpq*. In this model the *wpq* formula uses the values R = the number of seen representations, r_t = the number of seen representations containing term t, N = the number of representations in top-ranked documents, n_t = the number of representations containing the term t. This model uses an ostensive relevance profile to enhance the *wpq path model* presented in the previous section.

3.4 Random Term Selection Model

The random term selection model assigns a random score between 0 and 1 to terms from viewed representations. At the end of each relevance path, the model ranks the terms based on these random scores and uses the top-scoring terms to expand the original query. This model does not use any path weighting or indicativity measures. This model is a baseline and was included to test the degree to which using any reasonable term-weighting approach affected the success of the implicit feedback. Also, since it did not retain any memory of important terms or information objects this model was also expected to experience no learning.

3.5 Summary

We have introduced a variety of implicit feedback models based on binary voting, Jeffrey's rule of conditioning, three using *wpq* query expansion and random term selection. In this study we compare these models based on the degree to which each improves search effectiveness and learns relevance. In the next section we describe the searcher simulation that tests these models.

4 Simulation-Based Evaluation Methodology

There has been no precedent set on how to best evaluate implicit feedback models. In this study we use a simulation-based evaluation methodology to benchmark such models and choose the best performing models for future studies with real searchers.

The simulation assumes the role of a searcher, browsing the results of an initial retrieval. The information content of the top-ranked documents in the first retrieved document set constitutes the information space that the searcher must explore. All interaction in this simulation is with this set (i.e. we never generate a new information space) and we assume that searchers will only view relevant information (i.e. only follow relevance paths from relevant documents).

4.1 System, Corpus and Topics

We use the popular SMART search system [11] and index the San Jose Mercury News (SJMN 1991) document collection taken from the TREC initiative [13]. This collection comprises 90,257 documents, with an average 410.7 words per document (including document title), an average 55.6 relevant documents per topic and has been used successfully in previous experiments of this nature [9].

We used TREC topics 101-150 and took query from the short *title* field of the TREC topic description. For each query we use the top 30 documents to generate relevance paths for use in our simulation. Although the collection comes with a list of

50 topic (query) descriptions, we concentrate on those queries with relevant documents from which to generate relevance paths. We exclude those queries where there are no relevant documents in the top 30 documents retrieved and queries for which there were no relevant documents. We use 43 of the original 50 topics in our study.

4.2 Relevance Paths

Real searchers would typically follow a series of *related* relevance paths in a rational way, viewing only the most useful or interesting. In this study we try to simulate the searcher, but do not make such decisions. Instead, we select a set of paths from the large set of potential paths generated *at random* from top-ranked relevant documents.

Each relevant document has a number of possible relevance paths. In Table 1 we give all routes for all path types. Since we deal with granular representations of documents, we do not include the sixth and final *Document* step in these paths.

Table 1. Possible relevance path routes

TRS	Title	Summary	Summary Sentence	Sentence in Context	Total
4	1	1	4	1	**16**
4	1	1	4		**16**
4	1	1			**4**
4	1				**4**
4					**4**
	1	1	4	1	**4**
	1	1	4		**4**
	1	1			**1**
	1				**1**

For example, for viewing all five representations (first row of Table 1) there are $4 \times 1 \times 1 \times 4 \times 1 = 16$ possible paths. The final column shows the total for each possible route. There are 54 possible relevance paths for each document. If all top 30 documents are relevant there are 1,620 (54×30) possible relevance paths.

In our study we use only a subset of these possible paths. The simulation assumes that searchers interact with relevant information, and not with every possible relevance path. Even though it was possible to use all paths for each query, different queries have different numbers of relevant top-ranked documents (and hence possible relevance paths). For the sake of comparability and consistency, we only use a subset of these paths, chosen randomly from all possible. The subset size is constant for all models.

4.3 Relevant Distributions and Correlation Coefficients

A good implicit feedback model should, given evidence from relevant documents, learn the distribution across the relevant document set. The model should train itself, and become attuned to searcher needs in the fewest possible iterations.

A relevant term space for each topic is created before any experiments are run. This space contains terms from all the relevant documents for that topic, ordered

based on their probability of relevance for that topic, computed in the same way as Equation 4.

After each iteration we calculate the extent to which the term lists generated by the implicit model correlates with the relevant distribution. The simulation 'views' relevance paths from relevant documents and provides the models with the implicit relevance information they need to train themselves. We measure how well the models *learn* relevance based on how closely the term ordering they provide matches the term ordering in the relevant distribution.

To measure this we use two nonparametric correlation coefficients, *Spearman's rho* and *Kendall's tau.* These have equivalent underlying assumptions and statistical power, and both return a coefficient in the range [-1,1]. However, they have different interpretations; the Spearman accounts for the proportion of variability between *ranks* in the two lists, the Kendall represents the difference between the probability that the lists are in the same order versus the probability that the lists are in different orders. We used both coefficients to verify learning trends.

4.4 Evaluation Procedure

The simulation creates a set of relevance paths for all relevant documents in those top-ranked documents retrieved for each topic. It then follows a random-walk of *m* relevance paths, each path is regarded as a feedback *iteration* and *m* is chosen by the experimenter. After each iteration, we monitor the effect on search effectiveness and how closely the expansion terms generated by the model correlate with the term distribution across that topic's relevant documents. We use this correlation as a measure of how well the model learns the relevant term distribution and precision as a measure of search effectiveness.

The following procedure is used *for each topic with each model*:

i. use SMART to retrieve document set in response to query (i.e. topic title) using an *idf* weighting scheme

ii. identify relevant documents in the top 30 retrieved documents

iii. create a query-biased summary of all relevant documents from top 30 in parallel using the approach presented in [15]

iv. create and store all potential relevance paths for each relevant document (up to a maximum of 54 per document)

v. choose random set of *m* relevance paths (iterations) from those stored (using the Java[3] random number generator)

vi. for *each* of the *m* relevance paths:

   a. weight terms in path with chosen model

   b. monitor Kendall and Spearman by comparing order of terms with order in that relevant distribution for that topic

[3] http://java.sun.com

c. choose top-ranked terms and use them to expand original query

d. use new query to retrieve new set of documents

e. compute new precision and recall values

To better represent a searcher exploring the information space, all simulated interaction was with the results of the first retrieval only. All subsequent retrievals were to test the effectiveness of the new queries and were not used to generate relevance paths. In the next section we describe our study.

4.5 Study

In our study we test how well each model learned relevance and generated queries that enhanced search effectiveness. We ran the simulation ten times for each implicit model, over all 43 'useable' topics. We added six terms to the query, this was done without any prior knowledge of the effectiveness of adding this number of terms to queries for this collection. We set $m = 20$ and hence *each run comprised 20 iterations* (i.e. relevance paths or documents). We recorded correlation coefficients and measures of search effectiveness at iterations 1, 2, 5, 10 and 20. Using these iterations allowed us to monitor performance at different points in the search. In the document-centric approach each *document* is an iteration. Therefore, in this model, it was only possible to have as many iterations as there were relevant top-ranked documents.

5 Results

The study was conducted to evaluate a variety of implicit feedback models using searcher simulations. In this section we present results of our study. In particular we focus on results concerning search effectiveness and relevance learning. We use the terms *bvm*, *jeff*, *wpq.doc*, *wpq.path*, *wpq.ost* and *ran* to refer the binary voting, Jeffrey's, wpq document, wpq path, wpq ostensive and random models respectively.

5.1 Search Effectiveness

We measure search effectiveness for each of our implicit models through their effects on precision[4]. Figure 2 shows the 11pt precision values for each model across all iterations. As the figure illustrate, all models increased precision as the number of iterations increases.

Figure 2 presents the actual precision values across all 20 iterations. The Jeffrey's and binary voting models outperform the other implicit feedback models, with large increases inside the first five iterations. Both models are quick to respond to implicit relevance information, with the largest marginal increases (change from one iteration to the next) coming in the first iteration. The other models do not perform as well, but steadily increase until around 10 iterations where precision levels out.

[4] Both precision and recall were improved by the implicit models. However, since we only consider the top-30 documents the effects on precision are of more interest in this study.

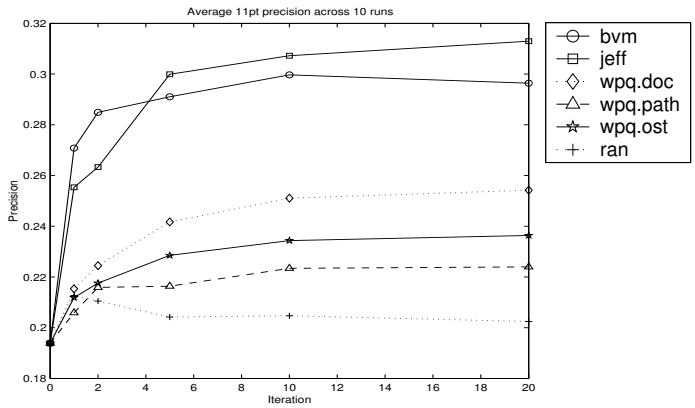


Fig 2. Average precision across 20 feedback iterations

Table 2 illustrates the marginal difference more clearly than Figure 2, showing the percentage change overall and the marginal percentage change at each iteration.

Table 2. Percentage change in precision per iteration. Overall change in first column, marginal change in second shaded column. Highest percentage in each column in bold

	Iterations									
Model	1		2		5		10		20	
bvm	**28.4**	–	**31.9**	**4.9**	33.4	2.9	35.3	2.9	34.6	–1.1
jeff	24.1	–	26.4	3.0	**35.3**	**12.2**	**36.9**	2.4	**38**	**1.8**
wpq.doc	10	–	13.6	4.1	19.8	7.1	22.8	**3.7**	23.7	1.2
wpq.path	5.8	–	10.2	4.6	10.4	0.2	13.2	3.2	13.4	0.2
wpq.ost	8.5	–	10.9	2.6	17.2	4.8	17.2	2.5	18	0.9
ran	8.8	–	7.9	–1.1	5	–3.1	5.3	0.2	4.2	–1.1

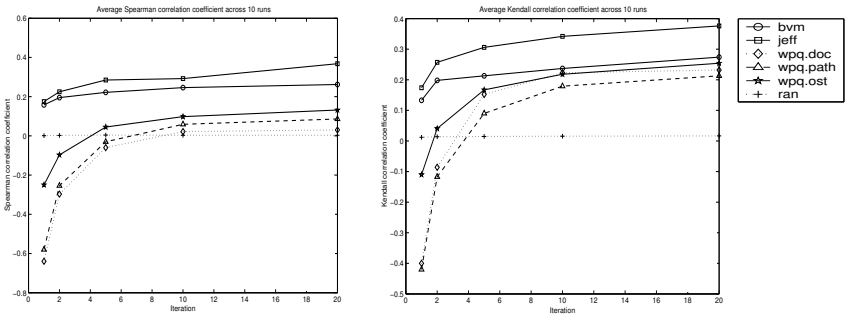
As Table 2 shows the largest increases in precision overall and marginally come from the binary voting model and the Jeffrey's model. Although after 20 iterations the marginal effects of all models appear slight. The random model performs poorly, although still leads to small overall increases in precision over the baseline. Even though the *random model* assigned each term a random score, the paths selected by the simulation were still query-relevant. Our results show that choosing terms randomly from relevance paths can help improve short queries to a small degree.

The *wpq*-based models appeared to follow a similar trend. At each iteration we carried out a one-way repeated measures ANOVA to compare all three *wpq*-based models and *t*-tests for pair-wise comparisons where appropriate. During the first two iterations, there were no significant differences (iteration 1: $F_{2,27} = 2.258$, $p = 0.12$, iteration 2: $F_{2,27} = 1.803$, $p = 0.18$) between the *wpq* models tested. ANOVAs across iterations 5, 10 and 20 suggested there were significant differences in precision between the three *wpq*-models. A series of *t*-tests revealed the *wpq document model* performed significantly better than both path-based *wpq* models (ostensive-path and

path) for iterations 5, 10 and 20 ($p < 0.05$). We could therefore posit that perhaps the relevance paths were not of sufficient size and did not contain a sufficient mixture of terms from which *wpq* could choose candidates for query expansion.

5.2 Relevance Learning

We measured how well the implicit models trained themselves when given relevance information by the simulation by using the degree of correlation between the ordered list of terms in the topic's relevant distribution and the ordered list of terms chosen by the implicit model. Figure 3 shows the average Spearman (a) and Kendall (b) correlation coefficients across all 43 topics.



a b

Fig 3. Correlation coefficients; a. Spearman b. Kendall

Both coefficients follow similar trends for all implicit models. Again the Jeffrey's and binary voting model learn at a faster rate, with the Jeffrey's performing best. The random model returns a coefficient value close to zero with both coefficients. In both cases a value of zero implies no correlation between the two lists, and this was to be expected if the model randomly ordered the term list. For all other models the coefficients tends to one, implying that the models were *learning* the relevant distribution from the given relevance information. Both the Jeffrey's model and the binary voting model obtain high degrees of correlation after the first iteration, whereas the *wpq* models need more *training* to reach a level where the terms they recommend appear to match those in the relevant distribution.

6 Discussion

The implicit feedback models evaluated in this paper *all* increased search effectiveness through query expansion. However, two models performed particularly well; that based on Jeffrey's conditioning and the binary voting model. Both models improved precision and developed lists of terms that were closely correlated to those of the relevant distribution.

Initially the Jeffrey's does not perform as well as the binary voting model. However, after five paths it creates more effective queries and from then on performs increasingly better than it. The Jeffrey's model uses prior evidence that is independent of the searcher's interaction. Initial decisions are made based on this prior evidence, and for the first few iterations it is reasonable to assume that this evidence still plays a

part in term selection. However, as more evidence is gathered from searcher interaction the terms selected by Jeffrey's conditioning improve.

An advantage of the binary voting model, and perhaps why it performs well in the initial stages is that it does not rely on any prior evidence, selecting terms based only on the representations viewed by the searcher. However, the lists of potential terms offered stagnates after 10 paths, since in the binary voting model the effect of the scoring is cumulative, the high-scoring, high-occurrence terms, obtain a higher score after only a few initial paths and cannot be succeeded by lower-ranked terms in later paths. This often means that the same query is presented in iterations 10 and 20.

The implicit feedback models learned relevance from the evidence provided to them by the simulation. This form of *reinforcement learning* [7], where the model was repeatedly shown examples of relevant information, allowed us to test how well each model trained itself to recognise relevance. From the six models tested, our findings showed that the Jeffrey's and binary voting models learned at the fastest rate. In the first few iterations those models based on *wpq* performed poorly, suggesting that these models need more training to reach an acceptable level of relevance recognition and that the Jeffrey's and binary voting models make a more efficient use of the relevance information presented to them.

We used linear regression and compared the *rate of learning* against *precision* for each of the six implicit feedback models. The results showed that for all models, the rate of learning (i.e. *Spearman's rho* and *Kendall's tau*) followed the same trend as precision (*all* $r^2 \geq .8856$ and *all* $T_{38} \geq 6.48$, $p \leq .05$). The rate in which the models learn relevance appears to match the rate in which they are able to improve search effectiveness.

For almost all iterations on all models, the marginal increases in precision and correlation reduce as more relevant information is presented. The models appear to reach a point of saturation at around 10 paths, where the benefits of showing 10 more paths (i.e. going to iteration 20) are only very slight and are perhaps outweighed by the costs of further interaction. It is perhaps at this point where searcher needs would be best served with a new injection of different information or explicit searcher involvement.

The results appear to be collection-independent. We re-ran the same experiment using the Wall Street Journal 1990-1992 collection instead of SJMN 1991. The findings mirrored those obtained in this study.

In the absence of a proper methodology for evaluating interactive retrieval approaches we introduced a novel simulation-based evaluation strategy. In this scheme we simulate searcher actions through a relevant set of document representations. However, a potential drawback of the searcher simulation proposed in this paper is that it does not consider the intentionality in interaction. A real searcher will view a series of information objects in rational way, depending on their information need. The simulation chooses paths *at random* from the top-ranked documents, and uses these paths to simulate interaction. At present the information need persists at the relevant document level (i.e. we choose paths from relevant documents), we posit that if the simulation catered for persistence in the interaction (i.e. relevance paths were traversed rationally) then the increases in search effectiveness and relevance learning would perhaps be even higher than those obtained.

7 Conclusions

In this paper we used searcher simulations to evaluate of a variety of implicit feedback models. The models under test are ostensive in nature and use the exploration of the information space and the viewing of information objects as an indication of relevance. We tested six models in total, all using an ostensive paradigm but each employing a different term selection stratagem.

We introduced implicit models based on Jeffrey's rule of conditioning, binary voting and three that use the popular *wpq* query expansion approach. The simulated approach used to test the model assumes the role of a searcher 'viewing' relevant documents and relevance paths between granular representations of documents. The simulation passes the information it viewed to the implicit models, which use this evidence of relevance to select terms to best describe this information. We investigated the degree to which each of the models improved search effectiveness and learned relevance. From the six models tested, the Jeffrey's model provided the highest levels of precision and the highest rate of learning.

The burden of explicitly providing relevance information in traditional relevance feedback systems makes implicit feedback an appealing alternative. Simulation experiments are a reasonable way to test the worth of implicit models such as those presented in this paper. These tests can ensure that only the most effective implicit models are chosen as potential substitutes for explicit RF in interactive information seeking environments. Implicit systems using the Jeffrey's model are under development.

Acknowledgements

The work reported is funded by the UK EPSRC, grant number GR/R74642/01.

References

1. Barry, C.L. 'Document Representations and Clues to Document Relevance'. *Journal of the American Society for Information Science*. 49. 14, 1293-1303. 1998.
2. Campbell, I. and van Rijsbergen, C.J. 'The ostensive model of developing information needs'. *Proceedings of the 3rd CoLIS Conference*, 251-268. 1996.
3. Hamming, R.W. 'Error-Detecting and Error-Correcting Codes', *Bell Systems Technical Journal*. 29. pp 147-160. 1950.
4. Harman, D. 'An Experimental Study of the Factors Important in Document Ranking'. In *Proceedings of the 9th ACM SIGIR Conference*, 186-193. 1986.
5. Jeffrey, R.C. *The Logic of Decision*, 2nd edition. University of Chicago Press. 1983.
6. Lam, W., Mukhopadhyay, S., Mostafa, J., and Palakal, M. 'Detection of Shifts in User Interests for Personalised Information Filtering'. *Proceedings of the 18th ACM SIGIR Conference,* 317-325. 1996.
7. Mitchell, T.M. *Machine Learning*. McGraw-Hill. 1997.
8. Robertson, S.E. 'On term selection for query expansion'. *Journal of Documentation*. 46. 4, 359-364. 1990.
9. Ruthven, I. 'Re-examining the Potential Effectiveness of Interactive Query Expansion'. *Proceedings of the 26th ACM SIGIR Conference*, 213-220. 2003.
10. Salton, G. (Ed.). *The SMART Retrieval System*. Prentice-Hall. 1971.
11. Salton, G. and Buckley, C. 'Improving retrieval performance by relevance feedback'. *Journal of the American Society for Information Science*. 41. 4. pp 288-297. 1990.
12. van Rijsbergen, C.J. 'Probabilistic Retrieval Revisited'. *The Computer Journal*. 35. 3, 291-298. 1992.
13. Voorhees, E.H. and Harman, D. 'Overview of the sixth text retrieval conference (TREC-6)'. *Information Processing and Management*. 36. 1, 3-35. 2000.
14. White, R.W., Jose, J.M. and Ruthven, I. The use of implicit evidence for relevance feedback in Web retrieval. *Proceedings of 24th ECIR Conference*, 93-109. 2002.
15. White, R.W., Jose, J.M. and Ruthven, I. A task-oriented study on the influencing effects of query-biased summarisation in web searching. *Information Processing and Management*. 39. 5, 707-733. 2003.
16. White, R.W., Jose, J.M. and Ruthven, I. An Approach for Implicitly Detecting Information Needs. *Proceedings of 12th CIKM Conference*, 504-508. 2003.