Pergamon

Information Processing & Management, Vol. 31, No. 4, pp. 605–620, 1995
Elsevier Science Ltd. Printed in Great Britain

0306-4573(95)00070-0

USER CHOICES: A NEW YARDSTICK FOR THE EVALUATION OF RANKING ALGORITHMS FOR INTERACTIVE QUERY EXPANSION

EFTHIMIS N. EFTHIMIADIS

Graduate School of Education and Information Studies, University of California at Los Angeles, 405 Hilgard Avenue, Los Angeles, CA 90024-1520, U.S.A.

(Received 7 October 1993; accepted in final form February 1994)

Abstract—The performance of eight ranking algorithms was evaluated with respect to their effectiveness in ranking terms for query expansion. The evaluation was conducted within an investigation of interactive query expansion and relevance feedback in a real operational environment. This study focuses on the identification of algorithms that most effectively take cognizance of user preferences. User choices (i.e. the terms selected by the searchers for the query expansion search) provided the yardstick for the evaluation of the eight ranking algorithms. This methodology introduces a user-oriented approach in evaluating ranking algorithms for query expansion in contrast to the standard, system-oriented approaches. Similarities in the performance of the eight algorithms and the ways that these algorithms rank terms were the main focus of this evaluation. The findings demonstrate that the *r-lohi*, *wpq*, *emim*, and *porter* algorithms have similar performance in bringing good terms to the top of a ranked list of terms for query expansion. However, further evaluation of the algorithms in different (e.g. full-text) environments is needed before these results can be generalized beyond the context of the present study.

1. INTRODUCTION

Information retrieval researchers have been advocating for more testing of probabilistic retrieval techniques in operational environments, because to date most experimentation on relevance feedback systems has been conducted in the laboratory (Sparck Jones, 1988). The research presented in this paper follows this line of thinking, and it is part of an investigation of interactive query expansion* (Efthimiadis, 1992). The overall research aim was to investigate the process of interactive query expansion (IQE) from various points of view, including effectiveness. In order to investigate the process of query expansion as well as its effectiveness, one needs to have a real system. Therefore, real users with their real requests were used in an operational environment, as opposed to searching a static test collection with fixed (artificial) queries, to study query expansion and relevance feedback in a dynamic, user-centered environment. For the research reported, the INSPEC database, on both Data-Star and ESA-IRS, was searched online using CIRT (Robertson *et al.*, 1986), a front-end system that allows weighting, ranking, and relevance feedback.

In the context of a real system, real requests, and real interaction, particular importance is given to the characteristics of the interaction. The selection of terms at a particular stage in the search process is, for example, the most obvious characteristic. Therefore, the results include both a quantitative and a qualitative component. In this research, data were collected from 25 searches. The data collection mechanisms included questionnaires, transaction logs, and relevance evaluations. The variables examined were divided into seven categories, which include: retrieval effectiveness, user effort, subjective user reactions, user characteristics, request characteristics, search process characteristics, and term selection characteristics.

*The terms interactive query expansion and semi-automatic query expansion are used interchangeably in the text.

Query expansion, as a process of supplementing the original query terms with additional terms, can be done automatically, manually, or interactively. Some of the research questions that emerge when considering query expansion are: What are good terms? Which are the best terms for query expansion? Where can we get the query expansion terms from? How useful can the query expansion terms be? How can we present the terms? How can we rank the terms? Which ranking algorithm or method should we use? Are searchers able to recognize the good terms? How do the searchers select terms? What criteria do searchers use? What kind of relationships are there between the original query terms and the terms that the users select? Is there a difference in what the user selects and what the system suggests as a good term?

The present study, like all operational system studies, has produced a wide range of results. With respect to query expansion and the associated questions outlined above, the most important results are reported from the term selection characteristics (Efthimiadis, submitted) and from the evaluation of the eight ranking algorithms. The latter is the focus of this paper. A review of automatic and semi-automatic query expansion is given elsewhere (Efthimiadis, 1992).

The research objectives of the study reported here were to: (a) investigate the behavior of the ranking algorithms for their effectiveness in ranking terms for interactive query expansion; (b) evaluate the proposed *r-lohi* algorithm (Efthimiadis, 1993) against existing algorithms; and (c) conduct the evaluation in the context of a real operational environment, as defined by CIRT. The evaluation focused on the characteristics of the different algorithms; how algorithms with similar performance treat terms for interactive query expansion; and how the ranking reflects the user choices, thus identifying the algorithms that most effectively take cognizance of user preferences.

2. TERM SELECTION FOR QUERY EXPANSION

In interactive query expansion, a term selection stage is required. At this stage, the system should present the query expansion terms to the user in some reasonable order. The order should preferably be one in which the terms most likely to be useful are close to the top of the list. During this stage, the system may also employ heuristic decisions to deal with the terms in more effective ways. For example, poor terms can be excluded from the term list instead of being given low weights.

In information retrieval, various algorithms have been proposed that attempt to quantify the value or usefulness of a query term in retrieval. Algorithms may estimate the term value based on some qualitative or quantitative criteria. The qualitative arguments are concerned with the "value" of the particular term in retrieval. The quantitative argument may involve some specific criterion, for example, a proof of performance such as the relevance weighting theory.

The relationship that holds between term frequency and term value and the effect on retrieval can be summarized as follows (Sparck Jones, 1971; Salton, 1975; van Rijsbergen, 1979):

- very frequent terms are not very useful;
- middle-frequency terms are quite useful;
- infrequent terms are likely to be useful, but not as much as the middle frequency terms;
- very infrequent terms are useful terms in the sense that when present they are good indicators of relevance. However, since these terms are not present most of the time, they do not help in retrieving very many documents.

From this knowledge it can be hypothesized that a good term ranking algorithm would bring the middle-frequency terms near the top of the list. Consequently, the evaluation of the ranking algorithms is based on this hypothesis.

3. RANKING ALGORITHMS

A plethora of ranking algorithms is reported in the literature (Sager & Lockemann, 1976; McGill *et al.*, 1979; Ro, 1988). Term weighting and ranking is discussed here as it relates to term

selection for query expansion, and only within the context of this investigation. The eight algorithms considered in this research (i.e. *f4*, *f4modified*, *porter*, *emim*, *wpq*, *zoom*, *r-lohi*, and *r-hilo*) are introduced below.

3.1. The f4 algorithm

The relevance weighting theory (Robertson & Sparck Jones, 1976) considers the use of relevance information as the basis for the weighting of query terms. It makes use of two kinds of assumptions: term independence assumptions and document ordering assumptions. The basic formula for the relevance weighting theory, which is also known as the binary independence retrieval model (BIM), is:

$$w_t = \log \frac{p_t(1-q_t)}{q_t(1-p_t)} \tag{1}$$

where p_t is the probability of term t occurring in a relevant document; and q_t is the probability of term t occurring in a nonrelevant document.

The application of this formula requires some knowledge as to the relative occurrence of the term in relevant or nonrelevant documents. The probability p and q may be estimated from relevance feedback information. So, given a sample of some, but not all, of the relevant documents, probability estimates can be made. In practice it is convenient to replace the probabilities by frequencies. To avoid infinite weights, when one of the estimates of eqn (1) is zero, a correction has been applied by adding 0.5 in each of the components of the equation. The result is known as the *f4* point-5 formula:

$$f4 = \log \frac{(r+0.5)(N-n-R+r+0.5)}{(n-r+r+0.5)(R-r+0.5)} \tag{2}$$

where the probability estimates are $p_t = r/R$ and $q_t = (n-r)/(N-R)$; and N is the total number of documents in the collection; R is the sample of relevant documents as defined by the user's feedback; n is the number of documents indexed by term t; r is the number of relevant documents (from the sample R) assigned to term t.

An account of the structure of relevance weights, the presence-absence components, and the approaches to estimation is given by Sparck Jones (1979, pp. 38–41). The matching function for each document is given by the '*simple sum of weights*' over all of the terms in the query.

3.2. The f4 modified algorithm

Robertson (1986) suggested a modification to the *f4* formula (referred to here as *f4modified*), which takes into consideration the addition of new terms to the original query. The modified formula is:

$$f4mod = \log \frac{(r+c)(N-n-R+r+1-c)}{(n-r+c)(R-r+1-c)} \tag{3}$$

where $c = n/N$ and r, R, n, N are the same as defined in the *f4*.

It was further suggested that the *f4modified* could be used in two ways (Robertson, 1986, p. 186). In automatic query expansion, every term from the relevant document would be weighted using $c = n/N$ and added to the search. In interactive query expansion, the terms would be weighted in the same fashion, and those selected by the user would then be weighted with the *f4* formula.

Among the aims of this research was to test the *f4modified* by using it as the weighting function of the query expansion terms. Apart from the theoretical argument for using the *f4modified*, there were not any empirical data available. Therefore, any first step was to compare

its behavior against that of the *f4* and look at what effect these formulae have on the ranking of the terms. A series of pilot tests using the *f4modified* gave rather unexpected results. The ranking of *f4modified* was almost identical to that of the *f4* (Efthimiadis, 1992). Both algorithms placed at the top of the ranked list terms that have low collection frequency (n) and also have low frequency in the relevant document set (r). This type of ranking, however, is explained by the nature of the relevance weighting theory, which assigns a higher degree of importance to the low-frequency terms, and therefore brings them to the top of the ranked list.

3.3. Porter's algorithm

Porter and Galpin (1988) in the MUSCAT online catalogue used the following ranking formula:

$$porter = \frac{r}{R} - \frac{n}{N} \tag{4}$$

where r, R, n, N are defined as in the *f4* weight [eqn (2)]. In the paper there is not any justification given for the formula, nor any explanation of how they arrived at it. However, upon looking at the formula, it can be established that the weight is influenced by a term's occurrence (r) in the relevant document set (R), as well as the term's frequency (n) in the collection (N). Since this formula is used for the ranking of terms for query expansion the:

r/R fraction: (a) never becomes 0, because in order to use the formula there must always be at least one document containing the term judged relevant, and (b) has a maximum value of 1; this happens whenever $r=R$. In other words, this portion of the weight can take values that fall within the range: $0<(r/R)\leq 1$;
n/N fraction: is influenced by a term's frequency (n). The higher the term frequency, the higher the result of the fraction.

Therefore, the *porter* formula seems to place more emphasis on terms that occur frequently in the relevant document set. This is because the *r/R* component dominates to such an extent that the *n/N* becomes noise and gets lost in the rounding (i.e. it is so small that its information content is lost).

3.4. The emim algorithm

The expected mutual information measure (*emim*) is a term weighting model incorporating relevance information in which it is assumed that index terms may not be distributed independently of each other (van Rijsbergen, 1977; Harper & van Rijsbergen, 1978; van Rijsbergen *et al.*, 1981).

$$emim = E_{iq} = \sum_{t_i, w_q} \Delta_{iq} P(t_i, w_q) \log \frac{P(t_i, w_q)}{P(t_i)P(w_q)} \tag{5}$$

or, more generally,

$$G_{iq} = \sum_{t_i, w_q} \Delta_{iq} D_{iq} P_{iq}$$

where t_i indicates the presence (1) or absence (0) of a term; w_q indicates that a document is relevant (1) or nonrelevant (0); Δ_{iq} indicates the value of a term as a relevance discriminator, and it is 1 if $t_i=w_q$ or -1 if $t_i\neq w_q$; D_{iq} is the "degree of involvement" (i.e. one of the four cells of the 2×2 contingency table); and P_{iq} is the "probabilistic contribution" given by the log expression.

The *emim* weight reduces to the *f4* weight when the "degree of involvement" (i.e. the joint

probabilities) are all unity. Assuming the same definitions for n, N, r, and R as those already used earlier, the *emim* weight of a term is calculated as follows:

$$E_{iq}=p_{11}i_{11}-p_{12}i_{12}-p_{21}i_{21}+p_{22}i_{22}$$

$$=\log\frac{rN}{Rn}\cdot r$$

$$-\log\frac{(n-r)N}{(N-R)n}\cdot(n-r)$$

$$-\log\frac{(R-r)N}{(N-n)R}\cdot(R-r)$$

$$+\log\frac{(N-n-R+r)N}{(N-n)(N-R)}\cdot(N-n-R+r)$$

3.5. The wpq algorithm

The results obtained from the empirical investigation of the *f4* and *f4modified* (i.e. the almost identical ranking) meant that some other algorithm based on the relevance weighting theory might offer a better alternative. As a result of the discussion regarding the behavior of these two algorithms, a new one was developed that was used for the empirical testing in this research (Robertson, 1990).

The independence assumption of the relevance weighting theory is that terms are distributed independently of each other in relevant documents, and also that terms are distributed independently of each other in nonrelevant documents. In the query expansion stage of a search, an additional assumption should be made that considers statistical independence between the query expansion term and the terms in the entire previous search formulation.

The distribution of the relevant items for the initial formulation, for example, is further divided into two distributions: one that describes the relevant items that contain the term and one that describes the relevant items that do not contain the term. These two new distributions are identical according to the independence assumption. In other words, the presence or absence of the query expansion term does not affect the initial distribution. When the entire collection is considered, these assumptions predict a positive association between an initial query formulation and a good new term for query expansion.

The inclusion of term t in the search formulation with weight w_t will increase the effectiveness of retrieval by

$$wpq=w_t(p_t-q_t) \tag{6}$$

where w_t is a weighting function, which in this case is the *f4*; p_t is the probability of term t occurring in a relevant document; and q_t is the probability of a term t occurring in a nonrelevant document.

This means that irrespective of the weighting function (w_t) used, the rule for deciding the inclusion of a term in a query expansion search should be based on the ranking of *wpq* instead of w_t alone. Substituting the weighting function and the probability of relevance in *wpq* with r, R, n, N, we get:

$$wpq=\log\frac{(r+0.5)(N-n-R+r+0.5)}{(n-r+0.5)(R-r+0.5)}\cdot\left(\frac{r}{R}-\frac{n-r}{N-R}\right) \tag{7}$$

The relevance weighting theory attempts, via the Probability Ranking Principle (Robertson, 1977), to optimize the entire length of the search curve from high-precision to high-recall. This

is expressed by the w_t component of the above formula, which assigns greater importance to the infrequent terms. However, a model that determines which term(s) to add for query expansion will have to lead to a preference somewhere between the very infrequent terms that lead to high precision and the frequent terms that lead to high recall.

In the above formula, this is achieved by $p_t - q_t$. This component, like the *porter* formula (4), is influenced by the frequency of occurrence of a term in the relevant document set (r), as well as the term's frequency (n) in the collection. Therefore, the multiplication of the two components results in the effect that seems to be required by the model for term selection in query expansion.

3.6. The zoom ranking

zoom was selected as one of the ranking algorithms to be studied because it is available in an online vendor system (i.e. ESA/IRS), and because it was used throughout this research for analyzing the relevant document sets.

zoom was first discussed in the literature by Martin (1982), and an analysis of it from a cognitive viewpoint was given by Ingwersen (1984). *zoom* is based on automatic frequency analysis of phrases, single words, codes, or a combination of these contained in a selected set of references. Once a set of records is generated in a file, the searcher may *zoom* the set. The *zoom* command can now analyze up to 20,000 records. However, the basic default analysis sample is 50 records across the last selected set.

The fields of the record that can be used to *zoom* vary from database to database. The searcher may request any of these fields, either on their own or in any combination. However, the *zoom* default fields are the controlled terms (CT) and uncontrolled terms (UT).

zoom processes the records in the set and the phrases and/or single words of the analysis are displayed in columns. All terms are ranked in descending order of their frequency of occurrence in the sample set. Within ties (i.e. whenever there are more than one term with the same frequency of occurrence), terms are ranked in alphabetical order. It is important to note that every occurrence of a term is counted. In addition, the analysis takes into account all terms including stopwords. However, all punctuation is removed in the display of the *zoom* text analysis.

3.7. The r-lohi algorithm

The *r-lohi* algorithm has been proposed by Efthimiadis (1993) as the result of the observation of the ranking behavior of algorithms used for ranking terms for query expansion.

The *r-lohi* ranking algorithm:

- ranks terms according to r, i.e. their frequency of occurrence in the relevant document set in descending order, and
- resolves ties according to their term frequency, n, from low-to-high frequency (hence, the abbreviation *r-lohi*, which is pronounced *r-low-high*).

It was hypothesized that the *r-lohi* algorithm would have a similar ranking to *porter* and a performance approaching that of *wpq* and *emim*.

One of the goals of the present study is to evaluate the performance of the proposed algorithm against that of the other algorithms, as reported in Efthimiadis (1993).

3.8. The r-hilo sort

A variant of the *r-lohi* algorithm is to rank candidate terms for query expansion using the *r-hilo* rank, which

- ranks terms according to r, that is, their frequency of occurrence in the relevant document set in descending order, and

- resolves ties according to their term frequency, n, from high-to-low frequency (abbreviated to *r-hilo*).

Since the *r-hilo* algorithm will result in sorting tied terms in exactly the opposite way of the *r-lohi* algorithm, it was included as a control for the study.

4. METHODOLOGY FOR EVALUATING RANKING ALGORITHMS FOR INTERACTIVE QUERY EXPANSION

To evaluate the effectiveness of a ranking algorithm for ranking terms for query expansion, IR experiments to date have focused on retrieval effectiveness. In other words, the effectiveness of ranking is measured through retrieval effectiveness expressed in terms of recall and precision. This means that an algorithm would be effective if during the experiment the top ranked terms used for query expansion will achieve high recall/precision levels, if tested only by itself in a test collection, or a higher recall level than some other algorithm, if it is a comparative test. This approach has been the focus for evaluating algorithms for automatic query expansion.

Interactive query expansion introduces new dimensions in the way algorithms could be evaluated. One such dimension comes from the users themselves, who are the ultimate judges of the performance of the system, and it is introduced here. Evaluation of the ranking algorithm was performed through the user selections of terms.

As mentioned earlier, an effective ranking algorithm will bring the good terms to the top of the list. Users, on the other hand, during the data collection of this research, studied the lists and identified all the useful terms as well as the five best terms. The user population consisted of faculty, researchers, and doctoral students. An example of a user request, a ranked list of terms, and the corresponding user choices are given in the Appendix.

The method employed for the evaluation was by assessing the distribution of the user-selected terms over the ranked list. Each list of terms was generated from the records identified as relevant to the query by each user during relevance feedback. Terms were extracted from the descriptor (DE) and identifier (ID) fields of these relevant documents. Users, therefore, were presented with lists of terms, and they provided relevance judgments for each term. They selected terms in two groups. One group contained all the terms they thought to be useful for the search. The other group contained the five best terms from those in the first group. These term choices were taken as the basis of the evaluation. In other words, given the user preferences for terms, how do the algorithms rank them? The user responses were matched against the ranking of the seven remaining algorithms.

The methodology that was followed is divided into three stages.

Stage 1. The methodology for the ranking of the terms of each search by the remaining seven algorithms was:

1. extract terms presented to users for each search ($N=25$);
2. calculate weights for the terms of every search with each of the seven algorithms (i.e. *f4, f4modified, r-lohi, emim, porter, r-hilo, zoom*);
3. divide each of the resulting ranked lists into (a) 2 parts (top half, bottom half), and (b) 3 parts (top third, middle third, bottom third);
4. match the user choices of terms to each ranked list;
5. for each list, tally the distribution of all the terms over each part, i.e. over (a) top half, bottom half, and (b) top third, middle third, bottom third;
6. for each list, tally the distribution of the user-designated five best terms over the top, middle, and bottom thirds.

Stage 2. The second stage of the evaluation of the eight ranking algorithms was to study the five top ranked terms of each list. The objective of this part of the evaluation was to look at these five top ranked terms and compare them in a qualitative manner. This type of analysis reports on the general impressions obtained about the terms. Furthermore, it provides additional explanation on the behavior of the algorithms, because the analysis focuses on the "best terms"

Table 1. Distribution of all the terms chosen by the users; ranked lists divided into two parts

Top half of list (all terms)						
Algorithm	*N*	Mean %	SD	SEMean	Min	Max
r-lohi	25	68	15	3	43	100
wpq	25	68	15	3	43	100
emim	25	68	16	3	43	100
porter	25	68	16	3	43	100
f4	25	65	19	4	36	100
f4mod	25	65	19	4	36	100
zoom	25	57	18	4	25	100
r-hilo	25	40	19	4	0	73

as identified by each algorithm.

Stage 3. The last part of the evaluation concentrated on the user-designated five best terms. These terms were identified by the users as being the best terms out of all the terms they checked as being useful, and were subsequently used for query expansion. Since these five terms are the most important for the users, they were studied in more depth. Therefore, the emphasis that each algorithm has placed on these terms was measured through the rank position of each of them.

The ranks of the terms were added, and the sum was used for comparisons. The rationale is that the sum of the ranks of the chosen terms would indicate the relative importance that each algorithm gives to the user preferences. Then, by comparing the differences between the sums of pairs of the algorithms, it can be established which algorithm comes closer to user preferences and whether there are any significant differences between the algorithms. In addition, correlation can be used as a measure of association between each pair of the algorithms. The following methodology was used:

1. Assign ranks (from 1 to N) to the terms in all ranked lists (6×25).
2. Establish rank position for each of the five best terms in each of the ranked lists.
3. Add the rank position information for the five terms of each list.
4. Use the Wilcoxon test to find the statistical significance on the performance of the algorithms.
5. Calculate the Pearson correlation coefficient r for pairs of algorithms.

The Wilcoxon test was chosen for the statistical analysis because this is a distribution-free test for matched pairs, and it is based on ranks (Lehmann, 1975). The Pearson product moment correlation coefficient, r, was used because it operates on each pair of data as they are, and not on ranks, as does Spearman's ρ. For each search there is a pair of values (the sum of the ranks) that correspond to two algorithms, and this information is maintained with Pearson's r.

The results for each of the three stages of the evaluation are discussed below.

5. RESULTS AND DISCUSSION

5.1. Distribution of the terms chosen by the users

This stage of the evaluation measures how well the algorithms have captured the terms selected by the users. These terms were identified by the users as good terms for query expansion. The greater the concentration of such terms at the top of a ranked list, the better the performance of the algorithm. In order to facilitate comparisons between the algorithms, the results have been presented in tabular format (Tables 1, 2 and 3), as well as in a bar chart (Fig. 1).

The distribution of the terms chosen by the users (as being potentially useful for each algorithm) over the lists, which are divided into two parts, are summarized, and statistics are

Table 2. Distribution of all the terms chosen by the users; ranked lists divided into three parts

Algorithm	*N*	Mean %	SD	SEMean	Min	Max
Top third (all terms)						
wpq	25	49	18	4	0	83
emim	25	49	19	4	0	86
r-lohi	25	49	20	4	0	100
porter	25	47	17	3	0	83
f4mod	25	46	20	4	0	86
f4	25	45	19	4	0	83
zoom	25	39	17	3	13	82
r-hilo	25	29	17	3	0	64
Middle third (all terms)						
wpq	25	31	17	3	9	100
emim	25	31	18	4	0	100
r-lohi	25	31	19	4	0	100
porter	25	32	17	3	0	100
f4mod	25	35	20	4	0	100
f4	25	35	20	4	11	100
zoom	25	33	13	3	11	67
r-hilo	25	28	18	4	0	100
Bottom third (all terms)						
wpq	25	19	12	2	0	43
emim	25	19	12	2	0	43
r-lohi	25	19	12	2	0	43
porter	25	19	12	2	0	43
f4mod	25	19	14	3	0	43
f4	25	19	14	3	0	43
zoom	25	27	15	3	0	63
r-hilo	25	42	19	4	0	83

Table 3. Distribution of all the five best terms chosen by the users; ranked lists divided into three parts

Algorithm	*N*	Mean %	SD	SEMean	Min	Max
Top third (5 terms)						
emim	25	64	25	5	0	100
wpq	25	63	24	5	0	100
r-lohi	25	62	27	5	0	100
porter	25	60	25	5	0	100
f4	25	54	25	5	0	100
f4mod	25	54	25	5	0	100
zoom	25	43	20	4	0	80
r-hilo	25	32	23	5	0	80
Middle third (5 terms)						
emim	25	22	24	5	0	100
wpq	25	23	24	5	0	100
r-lohi	25	24	26	5	0	100
porter	25	25	25	5	0	100
f4	25	31	23	5	0	100
f4mod	25	30	24	5	0	100
zoom	25	29	22	4	0	80
r-hilo	25	19	21	4	0	100
Bottom third (5 terms)						
emim	25	14	17	3	0	60
wpq	25	14	17	3	0	60
r-lohi	25	14	15	3	0	60
porter	25	15	17	3	0	60
f4	25	16	17	3	0	60
f4mod	25	16	17	3	0	60
zoom	25	28	21	4	0	75
r-hilo	25	50	24	5	0	100

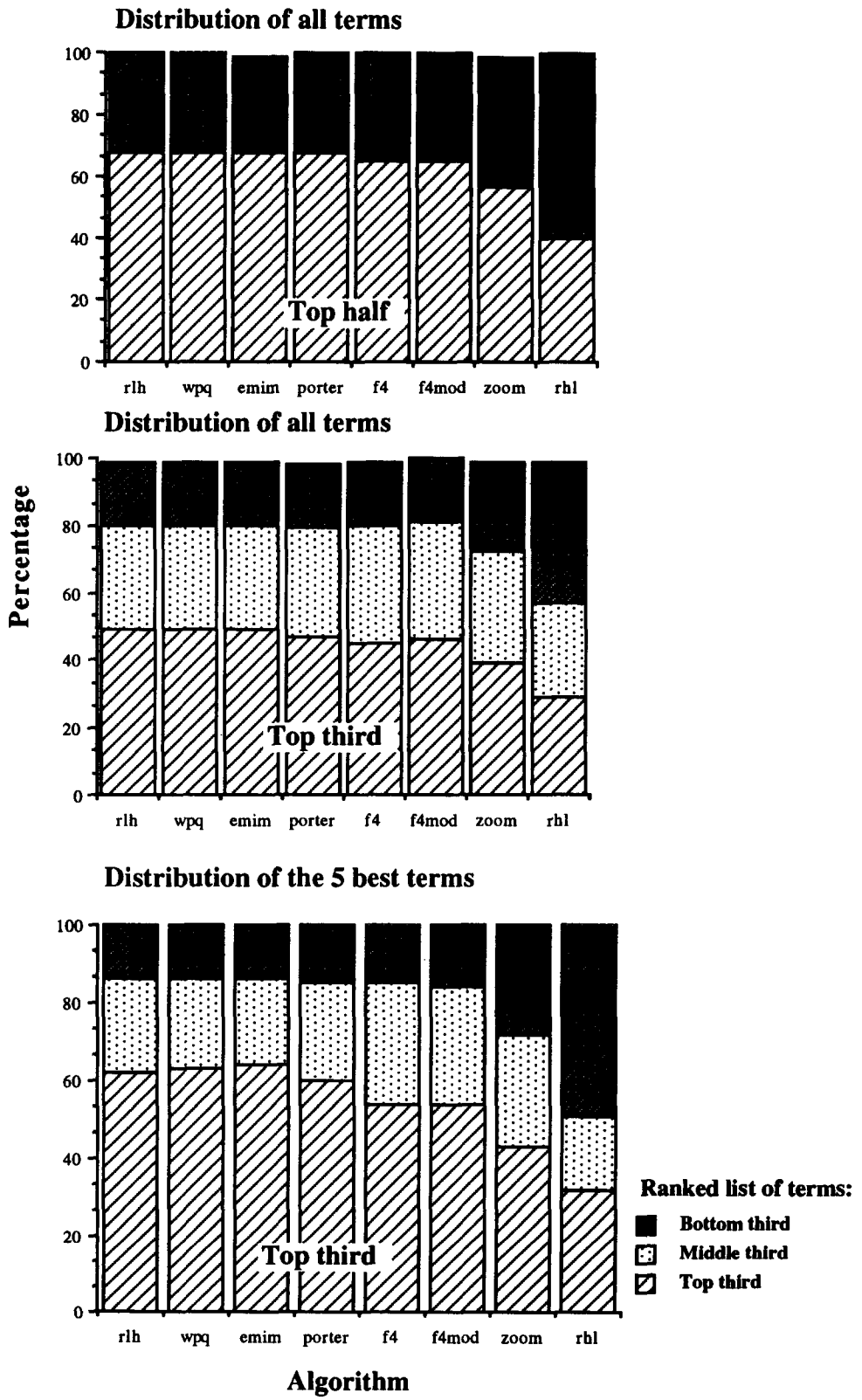

Fig. 1. Distribution of user-selected terms over the ranked lists for the eight algorithms (in percentages). Top and middle panels present the distribution of all terms when the ranked lists are divided into two and three parts, respectively. Bottom panel presents the distribution of the five best terms chosen by the users; ranked lists divided into three parts.

given in Table 1. This table presents for each of the eight algorithms the mean percentage value over the 25 searches for the top and the bottom halves. The data of Table 1 are also presented in Fig. 1. The concentration of terms at the top half ranged from as low as 40% for *r-hilo* and 57% for *zoom* to 68% for *wpq, emim, porter,* and *r-lohi. f4* and *f4modified* with 65% were not far behind the four top-rated algorithms.

The distribution of all the terms chosen by the users over the ranked lists, which are divided into three parts, are summarized in Table 2. This table gives for each algorithm the mean percentage values for the top, middle, and bottom thirds. The results presented in Table 2 are also illustrated by Fig. 1.

As seen in this figure, the three-way division of the ranked lists is more sensitive to the user preferences. It highlights the part of the list where the highest concentration is, and in this way it points out where each algorithm places emphasis on the list. The concentration of terms at the top third ranges from 29% for *r-hilo* and 39% for *zoom* to 49% for *wpq, emim* and *r-lohi*. In the middle third, the concentration ranges from 28% to 35%, and in the bottom third, the concentration of terms is 19% for all the algorithms except *zoom* and *r-hilo*, which are 27% and 42%, respectively.

From these results a pattern emerges for all algorithms except *zoom* and *r-hilo*. This is that the distribution of all the terms over the lists is on average proportional to 20%–30%–50% for the bottom, middle, and top thirds, whereas for *zoom* it is proportional to 30%–30%–40%, and for *r-hilo* it is proportional to 40%–30%–30%. Overall, the *wpq, emim,* and *r-lohi* algorithms, with 49%, have the highest concentration of terms at the top third.

Having looked at the distribution of all the terms, the distribution of the five best terms was then assessed. The distribution of the user-designated five best terms over the lists, which are divided into three parts, are summarized in Table 3. This table presents the mean percentage values of the term distribution in the top, middle, and bottom thirds for each algorithm.

The results presented in Table 3 are also given in Fig. 1. The breakdown of the user-identified five best terms, as seen in Fig. 1, provides a finer way of looking at the results. The figure highlights the part of the list where the terms are concentrated, and therefore brings out the differences between the algorithms.

The distribution of the five best terms is concentrated at the top third of the ranked list. The term concentration ranges from as low as 32% for *r-hilo* and 43% for *zoom*; to the highest concentration of 62%–64% for *r-lohi*, *wpq*, and *emim*.

From the results presented here, *r-lohi, wpq,* and *emim* emerge as the algorithms with the best performance. The algorithms have succeeded in capturing the majority of the user-preferred terms at the top of the ranked lists; 68% of all terms were at the top half of the lists, 49% of all terms were at the top third, and 62–64% of the five best terms were at the top third of the lists. This level of concentration of user-preferred terms at the top of a ranked list are acceptable for interactive query expansion.

5.2. The five top ranked terms of each algorithm

The five top ranked terms for each algorithm in every search were compiled in 25 tables. Each table was divided into eight smaller tables, one for every algorithm. For every term, its weight, term frequency (n), and frequency within the relevant document set (r) were given to facilitate comparisons.

The overall impressions from the study of the 25 tables are summarized below. From the lists the emerging pattern is that the terms between *wpq* and *emim, f4* and *f4modified*, and *porter* and *zoom* are very similar. That is, the ranking between these pairs of algorithms is very similar. The terms of the *r-lohi* algorithm are very similar to those of *wpq* and *emim* and less similar to *porter*. The *r-hilo* terms are closer to *zoom* than any other algorithm.

The observation of *wpq* and *emim* show that the top five positions contain almost the same terms throughout. However, the rank orders differ slightly. The terms from the *r-lohi* list as compared to the *wpq* and *emim* demonstrate very small differences, which are explained by the emphasis these algorithms put on n and r. The terms and the rank order found between *f4* and *f4modified* are so close as to be almost identical. *porter* and *zoom* give very similar terms and

rank positions in almost all searches. The *r-hilo* rankings are very similar to *zoom*. Differences in the rankings of these algorithms are due to the way ties are resolved. *zoom* sorts ties in alphabetical order and *r-hilo* breaks ties from high to low term frequency. Porter and Galpin (1988) have not given any information of how ties should be resolved. The processing programs used in the data collection and for the ranking of the lists sorted the weights in reverse numerical order, which apparently ranked the ties in reverse alphabetical order.

The most noticeable differences seem to appear between the terms of *f4* and *f4modified* and the terms of the remaining algorithms. Furthermore, there appears to be a marked distinction between the two groups. The differences are mainly due to factors that influence the ranking in each algorithm. *f4* and *f4modified* are influenced primarily by n, whereas the remaining algorithms are influenced by r. Overall, when r is not tied, *wpq*, *emim*, *r-lohi*, *porter*, *zoom*, and *r-hilo* would rank very similarly. Differences will start occurring when ties arise. The size of the set of relevant documents (R) from which the observed terms have come has a mean of five documents. Even though this size may seem small in practice, it is an acceptable sample size on which to base estimates for a relevance feedback search. More differences may occur, however, between the algorithms, if the size of the set of relevant documents gets larger.

5.3. *Sum of ranks of the user-designated five best terms*

Table 4(a) gives that sum of the ranks of the five best terms selected by the users for each search and for each algorithm. Columns represent the eight algorithms, and rows give the sum of the ranks for a search under each algorithm. A statistical summary that describes these data, averaged for each algorithm, is given in Table 4(b). The summary gives the mean, median, standard deviation, standard error of the mean, and the minimum and maximum value of sums of ranks for each algorithm. The best mean values are given by *wpq* and *emim*, which are followed by *r-lohi* and *porter*, *f4modified*, and *f4* and *zoom* and *r-hilo*, in this order.

The results shown in Table 4(a) were subjected to statistical analysis with the Wilcoxon test to determine whether there is any significant difference in the performance of the algorithms. The algorithms were taken in pair combinations and were analyzed with the Wilcoxon test using

Table 4(a). Sum of the ranks of the five best terms chosen by the users for each of the eight algorithms

Searcher	*wpq*	*emim*	*r-lohi*	*porter*	*f4*	*f4mod*	*zoom*	*r-hilo*
101	148	144	149	149	107	107	169	221
102	76	80	81	82	101	100	69	76
103	209	207	222	220	230	229	403	374
105	115	117	122	116	159	160	205	202
108	61	63	62	60	72	71	64	92
110	71	71	86	83	82	79	156	201
111	32	33	31	32	101	98	97	90
112	112	112	105	117	92	92	253	383
113	70	73	90	78	53	53	236	277
114	45	45	44	49	108	106	133	128
115	54	54	62	57	75	75	82	125
116	104	99	107	107	92	92	69	142
117	125	124	132	133	94	95	182	237
118	82	82	86	85	86	86	179	155
119	199	198	204	203	207	205	439	324
120	43	41	44	46	31	31	83	142
121	68	68	71	71	80	79	164	204
122	115	116	120	123	116	116	89	157
123	174	174	177	176	199	199	159	107
124	100	101	101	102	178	176	89	184
125	112	109	115	115	95	94	118	97
126	61	63	74	74	55	55	122	134
127	80	79	82	84	119	117	92	174
128	227	226	242	237	224	222	284	219
129	43	43	43	43	40	40	114	101

Table 4(b). Statistical summary of the sum of the ranks

Algorithm	N	Mean	Median	SDev	SEMean	Min	Max
wpq	25	101	82	54	11	32	227
emim	25	101	82	53	11	33	226
r-lohi	25	106	90	56	11	31	242
porter	25	106	85	55	11	32	237
f4	25	112	95	56	11	31	230
f4mod	25	111	95	56	11	31	229
zoom	25	162	133	98	20	64	439
r-hilo	25	182	157	85	17	76	383

the Minitab* Data Analysis Software, version 7. The results of the tests in terms of level of significance for each pair of algorithms are given in Table 5. From these results, it is established that pairs of algorithms that are significantly different in their performance are all combinations of either *zoom* or *r-hilo*. Pair combinations that were significantly different at the 5% level are: *f4* vs *zoom*, *f4mod* vs *zoom*, *zoom* vs *r-lohi*, and *porter* vs *zoom*. Combinations of pairs of algorithms that demonstrate highly significant difference in performance at the 1% level are: *wpq* vs *zoom*, *emim* vs *zoom*, *f4* vs *r-hilo*, *f4mod* vs *r-hilo*, *r-lohi* vs *r-hilo*, *porter* vs *r-hilo*, *wpq* vs *r-hilo*, and *emim* vs *r-hilo*. All other pair combinations of the algorithms show no significant difference in their performance (i.e. those with P value between 0.45 and 0.99, and are presented in the top part of Table 5).

To further analyze the data, the Pearson product moment correlation coefficient, r, calculated for pair combinations of the algorithms, is presented in Table 6. Correlation measures the strength of association between two variables. Therefore, the results in Table 6 show the strength of the relationship between the algorithms. Strong positive relationship is demonstrated by all pairs that have a value of $r>0.800$. The strongest association is found between *f4* and *f4modified* ($r=1.000$), *wpq* and *emim* ($r=0.999$), *wpq* and *porter* ($r=0.998$), *emim* and *porter* ($r=0.998$), *r-lohi* and *porter* ($r=0.997$), *r-lohi* and *emim* ($r=0.995$), and *r-lohi* and *wpq* ($r=0.994$).

6. CONCLUSIONS

The eight algorithms, *wpq*, *emim*, *r-lohi*, *porter*, *f4*, *f4mod*, *zoom*, *r-hilo*, were evaluated for their effectiveness in ranking terms for interactive query expansion.

Table 5. Levels of significance on pairs of algorithms for the Wilcoxon test

Algorithms	P value
Significant at the 5% level	
f4 vs *zoom*:	$P=0.033$
f4mod vs *zoom*:	$P=0.030$
zoom vs *r-lohi*:	$P=0.018$
porter vs *zoom*:	$P=0.017$
Significant at the 1% level	
wpq vs *zoom*:	$P=0.010$
emim vs *zoom*:	$P=0.0096$
f4 vs *r-hilo*:	$P=0.0014$
f4mod vs *r-hilo*:	$P=0.0012$
r-lohi vs *r-hilo*:	$P=0.0006$
porter vs *r-hilo*:	$P=0.0005$
wpq vs *r-hilo*:	$P=0.0003$
emim vs *r-hilo*:	$P=0.0002$

*Minitab is a registered trademark.

Table 6. Strength of association between algorithms as measured by Pearson's *r* correlation

Algorithm	*wpq*	*emim*	*f4*	*f4mod*	*porter*	*zoom*	*r-lohi*
emim	0.999						
f4	0.850	0.856					
f4mod	0.855	0.861	1.000				
porter	0.998	0.998	0.841	0.846			
zoom	0.729	0.733	0.613	0.615	0.737		
r-lohi	0.994	0.995	0.838	0.843	0.997	0.741	
r-hilo	0.558	0.559	0.397	0.400	0.574	0.830	0.564

In developing algorithms for the ranking of terms, in this case for the ranking of query expansion terms, the IR researcher focuses on the question of which might be the best terms to add during query expansion. What IR research tries to get from a ranking algorithm for that purpose is:

(a) to get the best terms at the top of the list; and
(b) to second-guess the user on which terms to choose.

These are the ideal goals that a ranking algorithm tries to achieve. However, these two goals are not necessarily compatible. For example, a ranking algorithm that treats terms according to some theoretical argument might not necessarily propose what the users will choose. It is the users who search, and we try to second-guess them. Therefore, this study has sought to identify the algorithms that most effectively take cognizance of user preferences by using the user choices of terms as the yardstick for the evaluation.

The methodology that was followed in measuring the distribution of terms chosen by the users, as seen in Fig. 1, is effective in demonstrating how the algorithms ranked the user preferences. Although the three-way division of the ranked lists is particularly sensitive to the user preferences, the user-identified five best terms provided a finer tool for the evaluation.

The results indicate that there appears to be very little difference in the ranking of *wpq*, *emim*, *r-lohi*, and *porter*. Similarly, there is almost no difference in the ranking of *f4* and *f4modified*. When comparing *wpq* and *emim* against *f4* and *f4modified*, there are major differences in the ordering of terms between the former pair of algorithms and the latter, but there is not any statistically significant difference in performance. *wpq* and *emim* are influenced by *r*, whereas *f4* and *f4modified* are dominated by *n*. The *r-lohi* and *porter*'s algorithm seem to have a very similar performance to that of *wpq* and *emim*, though the performance of *r-lohi* appears to be closer to that of *wpq* and *emim* than that of *porter*'s. The worst performance was achieved by *zoom* and *r-hilo*. The differences in performance reported here between the algorithms are explained by the way the algorithms resolve ties.

In conclusion, in the context of this evaluation the *wpq*, *emim*, *r-lohi*, and *porter* algorithms have outperformed all others in the ranking of the user-preferred terms for query expansion. The concentration of user-preferred terms at the top parts of the list achieved by these algorithms is high, 68% at the top half for all terms and 60–63% at the top third for the five best terms. Such concentration levels are probably acceptable for interactive query expansion because the user can browse the list and can recognize terms. However, for automatic query expansion, such a level of term concentration may not be acceptable, and it may indicate a need to further improve the performance of the algorithms.

Further evaluation in using automatic and interactive query expansion is needed before these findings can be generalized. Additional testing in both test collection and operational environments may bring out additional differences in performance of these algorithms that were not easily observed in the current study. For example, more differences between the algorithms may occur if the size of the set of relevant documents (*R*) is larger. Another point of concern for future research is how these algorithms would behave in a full-text environment. The present study used a bibliographic database (i.e. INSPEC). The descriptor and identifier fields were the source for the query expansion terms with an average length of 65 terms per ranked list. What would the difference be among the algorithms in a full-text environment where the average

length of a ranked list would be increased to hundreds or thousands of terms?

Finally, within the setup of the current study, the *r-lohi* algorithm has achieved very good performance in suggesting terms for query expansion. The *r-lohi* algorithm can be computed easily, and its major advantage is that it is independent of a particular retrieval technique. Therefore, it may be implemented in a variety of systems. For example, in a Boolean vendor system, such a Dialog, *r-lohi* may be used within Dialog's *RANK* and *TARGET* commands for the ranking of terms for query expansion.

REFERENCES

Efthimiadis, E. N. (1992). *Interactive query expansion and relevance feedback for document retrieval systems.* Unpublished doctoral dissertation, City University, London, UK.

Efthimiadis, E. N. (1993). A user-centered evaluation of ranking algorithms for interactive query expansion. In R. Korfhage, E. Rasmussen, and P. Willett (Eds), *Proceedings of the 16th International Conference of the Association of Computing Machinery,* Specialist Interest Group in Information Retrieval (pp. 146–159). Pittsburgh, Pa: ACM Press.

Efthimiadis, E. N. Interactive query expansion: A user-based evaluation in a relevance feedback environment. Manuscript submitted for publication.

Harper, D. J., & van Rijsbergen, C. J. (1978). An evaluation of feedback in document retrieval using co-occurrence data. *Journal of Documentation, 34*(3), 189–216.

Ingwersen, P. (1984). A cognitive view of three selected online search facilities. *Online Review, 8*(5), 465–492.

Lehmann, E. L. (1975). *Nonparametrics: Statistical methods based on ranks.* Oakland, Calif.: Holden Hay.

Martin, W. A. (1982). Helping the less experienced user. In D. Raitt (Ed.), *Proceedings of the 6th International Online Meeting,* London (pp.67–76). Oxford: Learned Information (Europe) Ltd.

McGill, M., Koll, M., & Noreault, T. (1979). An evaluation of factors affecting document ranking by information retrieval systems. Technical report, Syracuse University, School of Information Studies.

Porter, M. F., & Galpin, V. (1988). Relevance feedback in a public access catalogue for a research library: Muscat at the Scott Polar Research Institute. *Program, 22*(1), 1–20.

Ro, J. S. (1988). Evaluation of the applicability of ranking algorithms, Pt. I and Pt. II. *Journal of the American Society for Information Science, 39,* 73–78; *39,* 147–160.

Robertson, S. E. (1977). The probability ranking principle in IR. *Journal of Documentation, 33,* 294–304.

Robertson, S. E. (1986). On relevance weight estimation and query expansion. *Journal of Documentation, 42(3),* 182–188.

Robertson, S. E. (1990). On term selection for query expansion. *Journal of Documentation, 46*(4), 359–364.

Robertson, S. E., & Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science, 27*(3), 129–146.

Robertson, S. E., Bovey, J. D., Thompson, C. L., & Macaskill, M. J. (1986) Weighting, ranking and relevance feedback in a front-end system. *Journal of Information Science, 12,* 71–75.

Sager, W. K. H., & Lockemann, P. C. (1976). Classification of ranking algorithms. *International Forum for Information and Documentation, 1,* 12–25.

Salton, G. (1975). *Dynamic information and library processing.* Englewood Cliffs, N.J.: Prentice–Hall.

Sparck Jones, K. (Ed.) (1971). *Automatic keyword classification for information retrieval.* London: Butterworths.

Sparck Jones, K. (1979). Search term relevance weighting given little relevance information. *Journal of Documentation, 35*(1), 30–48.

Sparck Jones, K. (1988). A look back and a look forward. In Y. Chiaramella (Ed.), *Proceedings of the 11th International Conference on Research & Development in Information Retrieval,* Grenoble, France (pp. 13–29). Grenoble, France: IMAG and ACM Press.

van Rijsbergen, C. J. (1977). A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation, 33,* 106–119.

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

van Rijsbergen, C. J., Harper, D. J., & Porter, M. F. (1981). The selection of good search terms. *Information Processing & Management, 17(2),* 77–91.

APPENDIX

An example of an abbreviated user request, the initial query terms, the query expansion terms, and the ranked list of terms from which the query expansion terms were selected is given below.

Search topic

High level programming languages that are used to model information systems in a semantic level.

Initial query terms

programming language$1
concept$3
semantic$1
galileo

Query expansion terms
interactive conceptual language
high level language
classification
type hierarch$3

In each search, a ranked list of terms was presented to the user, who first selected the terms thought to be useful for the search and then ranked them in order of perceived importance. The ranked list below is divided in halves ($\frac{1}{2}$) and thirds ($\frac{1}{3}$) for the analysis as indicated in the text.

Ranked list of terms	User choices
galileo	
database management systems	
type hierarchies	4
abstraction mechanisms	
high level languages	2
semantic data model features	
interactive conceptual language	1
object oriented database language	8
modularization mechanism	
dialogo	7
flexible type system	
database designer s workbench	
programming language	
data structures	
	$\frac{1}{3}$
strongly typed programming language	
conceptual language	6
semantic integrity constraints	
database programming language	
high level interface	
interactive programming language	
	$\frac{1}{2}$
semantic data models	
semantic data model	
object oriented database	
database structure	
interactive environment	
database applications	
	$\frac{1}{3}$
database design	
query languages	
data types	
conceptual design	5
programming environments	
dbms	
aggregation	
interactive systems	
systems analysis	
generalization	
classification	3
database	
cad	