Full Text Retrieval based on Probabilistic Equations with Coefficients fitted by Logistic Regression

Wm. S. Cooper
Aitao Chen
Fredric C. Gey

S.L.I.S., University of California
Berkeley, CA 94720

ABSTRACT

The experiments described here are part of a research program whose objective is to develop a full-text retrieval methodology that is statistically sound and powerful, yet reasonably simple. The methodology is based on the use of a probabilistic model whose parameters are fitted empirically to a learning set of relevance judgements by logistic regression. The method was applied to the TIPSTER data with optimally relativized frequencies of occurrence of match stems as the regression variables. In a routing retrieval experiment, these were supplemented by other variables corresponding to sums of logodds associated with particular match stems.

Introduction

The full-text retrieval design problem is largely a problem in the combination of statistical evidence. With this as its premise, the Berkeley group has concentrated on the challenge of finding a statistical methodology for combining retrieval clues in as powerful a way as possible, consistent with reasonable analytic and computational simplicity. Thus our research focus has been on the general logic of how to combine clues, with no attempt made at this stage to exploit as many clues as possible. We feel that if a straightforward statistical methodology can be found that extracts a maximum of retrieval power from a few good clues, and the methodology is clearly hospitable to the introduction of further clues in future, progress will have been made.

We join Fuhr and Buckley (1991, 1992) in thinking that an especially promising path to such a methodology is to combine a probabilistic retrieval model with the techniques of statistical regression. Under this approach a probabilistic model is used to deduce the general form that the document-ranking equation should take, after which regression analysis is applied to obtain empirically-based values for the constants that appear in the equation. In this way the probabilistic theory is made to constrain the universe of logically possible retrieval rules that could be chosen, and the regression techniques complete the choice by optimizing the model's fit to the learning data.

The probabilistic model adopted by the Berkeley group is derived from a statistical assumption of 'linked dependence'. This assumption is weaker than the historic independence assumptions usually discussed. In its simplest form the Berkeley model also differs from most traditional models in that it is of 'Type 0' -- meaning that the analysis is carried out w.r.t. sets of query-document pairs rather than w.r.t. particular queries or particular documents. (For a fuller explanation of this typology see Robertson, Maron & Cooper 1982.) But when relevance judgement data specialized to the currently submitted search query is available, say in the form of relevance feedback or routing history data, the model is flexible enough to accommodate it (resulting in 'Type 2' retrieval.)

Logistic regression (see e.g. Hosmer & Lemeshow (1989)) is the most appropriate type of regression for this kind of IR prediction. Although standard multiple regression analysis has been used successfully by others in comparable circumstances (Fuhr & Buckley op. cit.), we believe logistic regression to be logically more appropriate for reasons set forth elsewhere (Cooper, Dabney & Gey 1992). Logistic regression, which accepts binary training data and yields probability estimates in the form of logodds values, goes hand-in-glove with a probabilistic IR model that is to be fitted to binary relevance judgement data and whose predictor variables are themselves logodds.

The Fundamental Equation

As the starting point of our probabilistic retrieval model we adopt (with reservations to be explained presently) a statistical simplifying assumption called the 'Assumption of Linked Dependence' (Cooper 1991). Intuitively, it states that in the universe of all query-document pairs, the degree of statistical dependency that exists between properties of pairs in the subset of all relevance-related pairs is linked in a certain way with the degree of dependency that exists between the same pair-properties in the subset of all nonrelevance-related pairs. Under at least one crude measure of degree of dependence (specifically, the ratio of the joint probability to the product of individual probabilities), the linkage in question is one of identity. That is, the claim made by the assumption is that the degree of the dependency is the same in both sets.

For N pair-properties $A_1, \ldots, A_N$, the mathematical statement of the assumption is as follows.

ASSUMPTION OF LINKED DEPENDENCE:

$$\frac{P(A_1, \ldots, A_N \mid R)}{P(A_1, \ldots, A_N \mid \bar{R})} = \frac{P(A_1 \mid R)}{P(A_1 \mid \bar{R})} \times \cdots \times \frac{P(A_N \mid R)}{P(A_N \mid \bar{R})}$$

If one thinks of a query and document drawn at random, R is the event that the document is relevant to the query, while each clue A_i is some respect in which the document is similar or dissimilar to, or somehow related to, the query.

For most clue-types likely to be of interest this simplifying assumption is at best only approximately true. Though not as implausible as the strong conditional independence assumptions traditionally discussed in probabilistic IR, still it should be regarded with skepticism. In practice, when the number N of clues to be combined becomes large the assumption can become highly unrealistic, with a distorting tendency that often causes the probability of relevance estimates produced by the resulting retrieval model to be grossly overstated for documents near the top of the output ranking. But it will be expedient here to adopt the assumption anyway, on the understanding that later on we shall have to find a way to curb its probability-inflating tendencies.

From the Assumption of Linked Dependence it is possible to derive the basic equation underlying the probabilistic model:

Equation (1):

$$\log O(R \mid A_1, \ldots, A_N) =$$

$$\log O(R) + \sum_{i=1}^{N} [\log O(R \mid A_i) - \log O(R)]$$

where for any events E and E' the odds $O(E \,/\, E')$ is by definition $\frac{P(E \,/\, E')}{P(\bar{E} \,/\, E')}$. Using this equation, the evidence of N separate clues can be combined as shown in the right side to yield the logodds of relevance based on all clues, shown on the left. Query-document pairs can be ranked by this logodds estimate, and a ranking of documents for a particular query by logodds is of course a probability ranking in the IR sense. For further discussion of the linked dependence assumption and a formal derivation of Eq. (1) from it, see Cooper (1991) and Cooper, Dabney & Gey (1992). An empirical investigation of independence and dependence assumptions in IR is reported by Gey (1993).

Application to Term Properties

We shall consider as a first application of the modeling equation the problem of how to exploit the properties of match terms. By a 'match' term we mean a stem (or word, phrase, etc.) that occurs in the query and also, in identical or related form, in the document to which the query is being compared. The retrieval properties of a match term could include its frequency characteristics in the query, in the document, or in the collection as a whole, its grammatical characteristics, the type or degree of 'match' involved, etc. If there are M match terms that relate a certain query to a certain document, Eq. (1) becomes applicable with N set to M. Each of the properties $A_1, \ldots, A_M$ then represents a composite clue or set of properties concerning one of the query-document pair's match terms.

Suppose a 'learning set' of human judgements of relevance or nonrelevance is available for a sample of representative query-document pairs. (However, for the time being we assume no such learning data is available for the very queries on which the retrieval is to be performed.) Logistic regression can be applied to the data to develop an equation capable of using the match term clues to estimate, for any query-document pair, values for each of the expressions in the right side of Eq. (1). Eq. (1) then yields a probability estimate for the pair.

TABLE I: Equations to Approximate the Components of Eq. (1)

$$
\begin{array}{rcll}
\log O(R) & \approx & & [\, b_0 + b_1 M] \\
\log O(R \mid A_1) - \log O(R) & \approx & [\, a_0 + a_1 X_{1,1} + \cdots + a_K X_{1,K} + a_{K+1} M \,] \quad - & [\, b_0 + b_1 M] \\
\cdot \qquad \cdot & \approx & \cdot & \cdot \\
\cdot \qquad \cdot & \approx & \cdot & \cdot \\
\cdot \qquad \cdot & \approx & \cdot & \cdot \\
\log O(R \mid A_M) - \log O(R) & \approx & [\, a_0 + a_1 X_{M,1} + \cdots + a_K X_{M,K} + a_{K+1} M \,] \quad - & [\, b_0 + b_1 M] \\
\hline
\log O(R \mid A_1 , \ldots , A_M) & \approx & [\, a_0 M + a_1 \sum_{m=1}^{M} X_{m,1} + \cdots + a_K \sum_{m=1}^{M} X_{m,K} + a_{K+1} M^2 \,] - (M-1)\,[\, b_0 + b_1 M \,] &
\end{array}
$$

What must be done, then, is to find approximating expressions for the various components in the right side of Eq. (1). Table I (above) shows the interrelationships among the quantities involved. In this array, the material to the left of the column of approximation signs just restates Eq. (1), for Eq. (1) asserts that the expressions above the horizontal line sum to the expression below the line. On the right of each approximation sign is a simple formula that might be used to approximate the expression on the left. (More complex formulae could of course also be considered.) Each right-side formula involves a linear combination of K variables $X_{m,1}, \ldots , X_{m,K}$ corresponding to the K retrieval properties (frequency values, etc.) used to characterize the term in question. The idea is to apply logistic regression to find the values for the coefficients (the a's and b's) in the right side that produce the best regression fit to the learning sample. Once these constants have been determined, retrieval can be carried out by summing the right sides to obtain the desired estimate of $\log O(R \mid A_1 , \ldots , A_N)$ for any given query-document pair.

It may not be immediately apparent why terms in M have been included in the approximating formulas on the right. In Eq. (1), the number N of available retrieval clues is actually part of the conditionalizing information, a fact that could have been stressed by writing $O(R \mid N)$ in place of $O(R)$ and $O(R \mid A_i, N)$ in place of $O(R \mid A_i)$. So the approximating formula for $\log O(R)$ is really an approximation for $\log O(R \mid M)$ and must involve a term in M. (Simple functions of M such as $\sqrt{M}$ or $\log (M)$ might also be considered.) Similar remarks apply to the formulas for approximating $\log (R \mid A_m)$.

There are two approaches to the problem of finding coefficients to use in the approximating expressions. The first is what might be called the 'triples-then-pairs' approach. It starts out with a logistic regression analysis performed directly on a sample of query-document-term triples constructed from the learning set. In the sample, each triple is accompanied by the clue-values associated with the match term in the query and document, and by the human relevance judgement for the query and document. The clue-values are used as the values of the independent variables in the regression, and the relevance judgements as the values of the the dependent variable. After the desired coefficients have been obtained, the resulting regression equation can be applied to evaluate the left-side expressions, which can in turn be summed down the M terms to obtain a preliminary estimate of $\log O(R \mid A_1, \ldots , A_M)$.

A second regression analysis is then performed on a sample of query-document pairs, using the preliminary logodds prediction from the first (triples) analysis as an independent variable. This second analysis serves the purpose of correcting for distortions introduced by the Assumption of Linked Dependence. It also allows retrieval properties not associated with particular match terms, such as document length, age, etc., to be introduced. The triples-then-pairs approach is the one used by the Berkeley group in its Trec-1 research (Cooper, Gey & Chen 1992). The theory behind it is presented in more detail in (Cooper, Dabney & Gey 1992).

The second way of determining the coefficients -- the one used in Trec-2 -- is the 'pairs-only' approach. It requires only one regression analysis, performed on a sample of query-document pairs. It is based on the trivial observation that in the right side of the above array, instead of adding across rows and then down the resulting column of sums, one can equivalently add down columns and across the resulting row of sums. Under either procedure the grand total value for log $O(R|\, A_1, \ldots, A_M)$ will be the same.

Summing down the columns and then across the totals gives the expression shown in the bottom line of the array. It simplifies to

$$\log O(R\,|A_1\,, \ldots, A_N) \;\approx$$

$$a_1 \sum_{m=1}^{M} X_{m,1} + \cdots + a_K \sum_{m=1}^{M} X_{m,K}$$
$$+ (a_0 - b_0 + b_1)M + (a_{K+1} - b_1)M^2$$

Since there is no need to keep the a_j coefficients segregated from the b_j coefficients to get a predictive equation, this suggests the adoption of a regression equation of form

$$\log O(R|\, A_1, \ldots, A_M) \;\approx$$

$$c_0 + c_1 \sum_{m=1}^{M} X_{m,1} + \cdots + c_K \sum_{m=1}^{M} X_{m,K}$$
$$+ c_{K+1} M \;+\; c_{K+2} M^2$$

The coefficients $c_0, \ldots, c_{K+2}$ may be found by a logistic regression on a sample of query-document pairs constructed from the learning sample. In the sample each pair is accompanied by its K different $X_{m,k}$-values each already summed over all match terms for the pair, the values of M and M^2, and (to serve as the dependent variable in the regression) the human relevance judgement for the pair.

But if only one level of regression analysis is to be performed, where is the correction for the Assumption of Linked Dependence to take place? That assumption causes mischief because it creates a tendency for the predicted logodds of relevance to increase roughly linearly with the number of match terms, whereas the true increase is less than linear. This can be corrected by modifying the variables in such a way that their values rise less rapidly than the number of match terms as the number of match terms increases. The variables can, for instance, be multiplied by some function $f(M)$ that drops gently with increasing M, say $\frac{1}{\sqrt{M}}$ or $\frac{1}{1 + \log M}$. The exact form of the function can be decided during the course of the regression analysis.

With such a damping factor included, the foregoing regression equation becomes

Equation (2):

$$\log O(R|\, A_1, \ldots, A_M) \;\approx$$

$$c_0 + c_1\; f(M) \sum_{1}^{M} X_{m,1} + \cdots + c_K\; f(M) \sum_{m=1}^{M} X_{m,K}$$
$$+ c_{K+1} M + c_{K+2} M^2$$

In our experiments, this simple modification was found to improve the regression fit and the precision/recall performance. It would appear therefore to be a worth-while refinement of the basic model. Note, however, that this adjustment only removes a general bias. It does nothing to exploit the possibility of measuring dependencies between particular stems to improve retrieval effectiveness. To exploit individual dependencies would be desirable in principle, but would require a substantial elaboration of the model for what might well turn out to be an insignificant improvement in effectiveness (for discussion see Cooper (1991)).

Optimally Relativized Frequencies

The philosophy of the project called for the use of a few well-chosen retrieval clues. The most obvious clues to be exploited in connection with match terms are their frequencies of occurrence in the query and document. What is not so obvious is the exact form the frequencies should take. For instance, should they be absolute or relative frequencies, or something in between?

The IR literature mentions two ways in which frequencies might be relativized for use in retrieval. The first is to divide the absolute frequency of occurrence of the term in the query or document by the length of the query or document, or some parameter closely associated with length. The second is to divide the relative frequency so obtained by the relative frequency of occurrence of the term in the entire collection considered as one long running text. Both kinds of relativization seem potentially beneficial, but the question remains whether these relativizations, if they are indeed helpful, should be carried out in full strength, or whether some sort of blend of absolute and relative frequencies might serve better.

To answer this question, regression techniques were used in a side investigation to discover the optimal extent to which each of the two kinds of relativization might best be introduced. To investigate the first kind of relativization -- relativization with respect to query or document length -- a semi-relativized stem frequency variable of the form

$$\frac{absolute\ \ frequency}{document\ length\ +\ C}$$

was adopted. If the constant C is chosen to be zero, one has full relativization, whereas a value for C much greater than the document length will cause the variable to behave in the regression analysis as though it were an absolute frequency. Several logistic regressions were run to discover by trial-and-error the value of C that produced the best regression fit to the TIPSTER learning data and the highest precision and recall. It was found that a value of around $C = 35$ for queries, and $C = 80$ for documents, optimized performance.

For the relativization with respect to the global relative frequency in the collection, the relativizing effect of the denominator was weakened by a slightly different method -- by raising it to a power less than 1.0. For the document frequencies, for instance, a variable of form

$$\frac{\dfrac{absolute\ \ frequency\ in\ doc}{document\ length\ +\ 80}}{[\ relative\ \ frequency\ in\ collection\ ]^{\ D}}$$

with $D < 1$ was in effect used. Actually, it was the logarithm of this expression that was ultimately adopted, which allowed the variable to be broken up into a difference of two logarithmic expressions. The optimal value of the power D was therefore obtainable in a single regression as the coefficient of the logged denominator.

Thus the variables ultimately adopted consisted in essence of sums over two optimally relativized frequencies ('ORF's) -- one for match stem frequency in the query and one for match stem frequency in the document. Because of the logging this breaks up mathematically into three variables. A final logistic regression using sums over these variables as indicated in Eq. (2) produced the ranking equation

Equation (3):

$$\log O(R|\ A_1, \ldots, A_M)\ \ \approx$$

$$-3.51\ +\ \frac{1}{\sqrt{M} + 1}\ \ \Phi\ +\ 0.0929\ M$$

where Φ is the expression

$$37.4\sum_{m=1}^{M} X_{m,1} + 0.330\sum_{m=1}^{M} X_{m,2} - 0.1937\sum_{m=1}^{M} X_{m,3}$$

Here

$X_{m,1}$ = number of times the m'th stem occurs in the query, divided by (total number of all stem occurrences in query + 35);

$X_{m,2}$ = number of times the m'th stem occurs in the document, divided by (total number of all stem occurrences in document + 80), quotient logged;

$X_{m,3}$ = number of times the m'th stem occurs in the collection, divided by the total number of all stem occurrences in the collection, quotient logged;

M = number of distinct stems common to both query and document.

Although Eq. (2) calls for an M^2 term as well, such a term was found not to make a statistically significant contribution and so was eliminated.

Eq. (3) provided the ranking rule used in the ad hoc run (labeled 'Brkly3') and the first routing run ('Brkly4') submitted for Trec-2. The equation is notable for the sparsity of the information it uses. Essentially it exploits only two ORF values for each match stem, one for the stem's frequency in the query and the other for its frequency in the document. Other variables were tried including the inverse document frequency (both logged and unlogged), a variable consisting of a count of all two-stem phrase matches, and several variables for measuring the tendency of the match stems to bunch together in the document. All of these exhibited predictive power when used in isolation, but were discarded because in the presence of the ORF's none produced any detectable improvement in the regression fit or the precision/recall performance. Some attempts at query expansion using the WordNet thesaurus also failed to produce noticable improvement, even when care was taken to create separate variables with separate coefficients for the synonym-match counts as opposed to the exact-match counts.

The quality of a retrieval output ranking matters most near the top of the ranking where it is likely to affect the most users, and matters hardly at all far down the ranking where hardly any users are apt to search. Because of this it is desirable to adopt a sampling methodology that produces an especially good regression fit to the sample data for documents with relatively high predicted probabilities of relevance. In an attempt to achieve this emphasis, the regression analysis was carried out in two steps. A logistic regression was first performed on a stratified sample of about one-third of the then-available TIPSTER relevance judgements to obtain a preliminary ranking rule. This preliminary equation was then applied as a screening rule to the entire available body of judged pairs, and for each query all but the highest-ranked 500 documents were discarded. The resulting set of 50,000 judged query-document pairs (100 topics $\times$ 500 top-ranked docs per topic) served as the learning sample data for the final regression equation displayed above as Eq. (3).

Application to Query-Specific Learning Data

To this point it has been assumed that relevance judgement data is available for a sample of queries typical of the future queries for which retrieval is to be performed, but not for those very queries themselves. We consider next the contrasting situation in which the learning data that is available is a set of past relevance judgements for the very queries for which new documents are to be retrieved. Such data is often available for a routing or SDI request for which relevance feedback has been collected in the past, or for a retrospective feedback search that has already been under way long enough for some feedback to have been obtained.

For a query so equipped with its very own learning sample, it is possible to gather separate data about each individual term in the query. Such data reflects the term's special retrieval characteristics in the context of that query. For instance, for a query term T_i we may count the number $n(T_i, R)$ of documents in the sample that contain the term and are also relevant to the query, the number of documents $n(T_i, \bar{R})$ that contain the term but are not relevant to the query, and so forth.

The odds that a future document will turn out to be be relevant to the query, given that it contains the term, can be estimated crudely as

$$O(R \mid T_i) \approx \frac{n(T_i, R)}{n(T_i, \bar{R})}$$

To refine this estimate, a Bayesian trick adapted from Good (1965) is useful. One simply adds the two figures used in expressing the prior odds of relevance (i.e. $n(R)$ and $n(\bar{R})$) into the numerator and denominator respectively with an arbitrary weight β as follows:

$$O(R \mid T_i) \approx \frac{n(T_i, R) + \beta\, n(R)}{n(T_i, \bar{R}) + \beta\, n(\bar{R})}$$

The smaller the value assigned to β, the more influence the sample evidence will have in moving the estimate away from the prior odds. The adjustment causes large samples to have a greater effect than small, as seems natural. It also forestalls certain absurdities implicit in the unadjusted formula, for instance the possibility of calculating infinite odds estimates.

Suppose now that a query contains Q distinct terms $T_1, \ldots, T_M, T_{M+1}, \ldots, T_Q$, numbered in such a way that the first M of them are match terms that occur in the document against which the query is to be compared. The fundamental equation can be applied by taking N to be Q, and interpreting the retrieval clues $A_1, \ldots, A_N$ as the presence or absence of particular query terms in the document. The pertinent data can be organized as shown in Table II (next page).

Thus we are led to postulate, for query-specific learning data, a retrieval equation of form

Equation (4):

$$\log O(R \mid A_1, \ldots, A_Q) \approx$$

$$c_0 + c_1 f(Q) \left[\Phi_1 + \Phi_2 - \Phi_3 \right]$$

where

$$\Phi_1 = \sum_{m=1}^{M} \log \frac{n(T_i, R) + \beta\, n(R)}{n(T_i, \bar{R}) + \beta\, n(\bar{R})}$$

$$\Phi_2 = \sum_{m=M+1}^{Q} \log \frac{n(\overline{T}_i, R) + \beta\, n(R)}{n(\overline{T}_i, \bar{R}) + \beta\, n(\bar{R})}$$

$$\Phi_3 = (Q-1) \log \frac{n(R)}{n(\bar{R})}$$

and where $f(Q)$ is some restraining function intended as before to subdue the inflationary effect of the Assumption of Linked Dependence. The coefficients c_0, c_1 are found by a logistic regression over a sample of query-document pairs involving many queries.

Combining Query-Specific and Query-Nonspecific Data

If a query-specific set of relevance judgements is available for the query to be processed, a larger

TABLE II: Reinterpretation of the Components of Eq. (1) for Routing Retrieval

$$
\begin{array}{rcccccc}
\log O(R) & \approx & & & & & \log \dfrac{n(R)}{n(\bar{R})} \\[2ex]
\log O(R \mid A_1) - \log O(R) & \approx & \log \dfrac{n(T_1, R) + \beta\, n(R)}{n(T_1, \bar{R}) + \beta\, n(\bar{R})} & & & - & \log \dfrac{n(R)}{n(\bar{R})} \\
\cdot \qquad \cdot & \approx & \cdot & & & & \cdot \\
\cdot \qquad \cdot & \approx & \cdot & & & & \cdot \\
\cdot \qquad \cdot & \approx & \cdot & & & & \cdot \\
\log O(R \mid A_M) - \log O(R) & \approx & \log \dfrac{n(T_M, R) + \beta\, n(R)}{n(T_M, \bar{R}) + \beta\, n(\bar{R})} & & & - & \log \dfrac{n(R)}{n(\bar{R})} \\[2ex]
\log O(R \mid A_{M+1}) - \log O(R) & \approx & & & \log \dfrac{n(\bar{T}_{M+1}, R) + \beta\, n(R)}{n(\bar{T}_{M+1}, \bar{R}) + \beta\, n(\bar{R})} & - & \log \dfrac{n(R)}{n(\bar{R})} \\
\cdot & \approx & & & \cdot & & \cdot \\
\cdot \qquad \cdot & \approx & & & \cdot & & \cdot \\
\cdot \qquad \cdot & \approx & & & \cdot & & \cdot \\
\log O(R \mid A_Q) - \log O(R) & \approx & & & \log \dfrac{n(\bar{T}_Q, R) + \beta\, n(R)}{n(\bar{T}_Q, \bar{R}) + \beta\, n(\bar{R})} & - & \log \dfrac{n(R)}{n(\bar{R})} \\
\hline
\log O(R \mid A_1, \ldots, A_Q) & \approx & \displaystyle\sum_{1}^{M} \log \dfrac{n(T_i, R) + \beta\, n(R)}{n(T_i, \bar{R}) + \beta\, n(\bar{R})} & + & \displaystyle\sum_{M+1}^{Q} \log \dfrac{n(\bar{T}_i, R) + \beta\, n(R)}{n(\bar{T}_i, \bar{R}) + \beta\, n(\bar{R})} & - & (Q-1) \log \dfrac{n(R)}{n(\bar{R})}
\end{array}
$$

nonspecific set may well be available at the same time. If so the theory developed in the foregoing section can be applied in conjunction with the earlier theory to capture the benefits of both kinds of learning sets. The retrieval equation will then contain variables not only of the kind occurring in Eq. (4) but also of the Eq. (2) kind.

It is convenient to formulate this equation in such a way that it contains as one of its terms the entire ranking expression developed earlier for the nonspecific learning data. For the TIPSTER data the combined equation takes the form:

Equation (5):

$$
\log O(R \mid A_1, \ldots, A_M, A'_1, \ldots, A'_Q) \approx
$$
$$
0.688\ \Phi_4 + 0.344 \left[\Phi_1 + \Phi_2 - \Phi_3 \right] + 0.0623
$$

where Φ_4 is the entire right side of Eq. (3) and Φ_1, Φ_2, Φ_3 are as defined in Eq. (4). This form for the equation is computationally convenient if Eq. (3) is to be used as a preliminary screening rule to eliminate unpromising documents, with Eq. (5) in its entirety applied only to rank those that survive the screening.

Eq. (5) was used to produce the Trec-2 routing run 'Brkly5'. Its coefficients were determined by a logistic regression constrained in such a way as to make the query-specific variables contribute about twice as heavily as the nonspecific, when contribution is measured by standardized coefficient size. This emphasis was largely arbitrary; finding the optimal balance between the query-specific and the general contributions remains a topic for future research. A value of $20 / n(\bar{R})$ was used for β. This choice too was arbitrary, and it would be interesting to try to optimize it experimentally for some typical collection (trying out, perhaps, numbers larger than 20, divided by the total number of documents in the query's learning set). No restraining function $f(Q)$ was used in the final form of Eq. (5) because none that were tried out produced any discernible improvement in fit or retrieval effectiveness in this context.

Calibration of Probability Estimates

The most important role of the relevance probability estimates calculated by a probabilistic IR system is to rank the output documents in as effective a search order as possible. For this ranking function it is only the relative sizes of the probability estimates that matter, not their absolute magnitudes. However, it is also desirable that the absolute sizes of these estimates be at least somewhat realistic. If they are, they can be displayed to provide guidance to the users in their decisions as to when to stop searching down the ranking. This capability is a potentially important side benefit of the probabilistic approach.

One way of testing the realism of the probability estimates is to see whether they are 'well-calibrated'. Good calibration means that when a large number of probability estimates whose magnitudes happen to fall in a certain small range are examined, the proportion of the trials in question with positive outcomes also falls in or close to that range. To test the calibration of the probability predictions produced by the Berkeley approach, the 50,000 query-document pairs in the ad hoc entry Brkly3 along with their accompanying relevance probability estimates were sorted in descending order of magnitude of estimate. Pairs for which human judgements of relevance-relatedness were unavailable were discarded; this left 22,352 sorted pairs for which both the system's probability estimates of relevance and the 'correct' binary judgements of relevance were available. This shorter list was divided into blocks of 1,000 pairs each -- the thousand pairs with the highest probability estimates, the thousand with the next highest, and so forth. Within each block the 'actual' probability was estimated as the proportion of the 1,000 pairs that had been judged to be relevance-related by humans. This was compared against the mean of all the system probability estimates in the block. For a well-calibrated system these figures should be approximately equal.

The results of the comparison are displayed in Table III. It can be seen that the system's probability predictions, while not wildly inaccurate, are generally somewhat higher than the actual proportions of relevant pairs. The same phenomenon of mild overestimation was observed when the runs Brkly4 and Brkly5 were tested for well-calibratedness in a similar way.

Since no systematic overestimation was observed when the calibration of the formula was originally tested against the learning data, it seems likely that the overestimation seen in the table is due mainly to the shift from learning data to test data. Naturally, predictive formulae that have been fine-tuned to a certain set of learning data will perform less well when applied to a new set of data to which they have not been fine-tuned. If this is indeed the root cause of the observed overestimation, it could perhaps be compensated for (at least to an extent sufficient for practical purposes) by the crude expedient of lowering all predicted probabilities to, say, around four fifths of their originally calculated values before displaying them to the users.

TABLE III: Calibration of Ad Hoc Relevance-Probability Estimates

Query-doc Pair Ranks	Mean System Probability Estimate	Proportion Actually Relevant
1 to 1,000	0.66	0.60
1,001 to 2,000	0.63	0.47
2,001 to 3,000	0.61	0.44
3,001 to 4,000	0.58	0.41
4,001 to 5,000	0.55	0.38
5,001 to 6,000	0.53	0.34
6,001 to 7,000	0.50	0.36
7,001 to 8,000	0.48	0.36
8,001 to 9,000	0.46	0.36
9,001 to 10,000	0.44	0.38
10,001 to 11,000	0.42	0.39
11,001 to 12,000	0.41	0.36
12,001 to 13,000	0.39	0.37
13,001 to 14,000	0.37	0.36
14,001 to 15,000	0.36	0.35
15,001 to 16,000	0.34	0.31
16,001 to 17,000	0.32	0.29
17,001 to 18,000	0.31	0.28
18,001 to 19,000	0.29	0.23
19,001 to 20,000	0.28	0.22
20,001 to 21,000	0.25	0.21
21,001 to 22,000	0.23	0.23
22,001 to 22,352	0.18	0.19

Computational Experience

The statistical program packages used in the course of the analysis included SAS, S, and BLSS. Of these, SAS provides the most complete built-in diagnostics for logistic regression. BLSS was found to be especially convenient for interactive use in a UNIX environment, and ended up being the most heavily used.

The prototype retrieval system itself was implemented as a modification of the SMART system with SMART's vector-similarity subroutines replaced by the probabilistic computations of Eqs. (3) and (5). For the runs Brkly3 and Brkly4, which used only Eq. (3), only minimal modifications of SMART were needed, and at run time the retrieval efficiency remained essentially the same as for the unmodified SMART system. This demonstrates that probabilistic retrieval need be no more cumbersome computationally than the vector processing alternatives. For Brkly5, which used Eq. (5), the modifications were somewhat more extensive and retrieval took about 20% longer.

Retrieval Effectiveness

The Berkeley system achieved an average precision over all documents (an '11-point average') of 32.7% for the ad hoc retrieval run Brkly3, and 29.0% and 35.4% respectively for the routing runs Brkly4 and Brkly5. The distinct improvement in effectiveness of Brkly5 over Brkly4 suggests that in routing retrieval the use of frequency information about individual query stems is worth while.

At the '0 recall level' a precision of 84.7%, the highest recorded at the conference, was achieved in the ad hoc run. The high effectiveness of the Berkeley system for the first few retrieved documents may be explainable in terms of the practice, mentioned earlier, of redoing the regression analysis for the highest-ranked 500 documents for each query. This technique ensures an especially good regression fit for the query-document pairs that are especially likely to be relevant, thus emphasizing good performance near the top of the ranking where it is most important.

The generally high retrieval effectiveness of the Berkeley system should be interpreted in the light of the fact that the system probably uses less evidence -- that is, fewer retrieval clues -- than any of the other high-performing TREC-2 systems. In fact, the only clues used were the frequency characteristics of single stems (not even phrases were included). What this suggests is that the underlying probabilistic logic may have the capacity to exploit exceptionally fully whatever clues may be available.

Summary and Conclusions

The Berkeley design approach is based on a probabilistic model derived from the linked dependence assumption. The variables of the probability-ranking retrieval equation and their coefficients are determined by logistic regression on a judgement sample. Though the model is hospitable to the utilization of other kinds of evidence, in this particular investigation the only variables used were optimally relativized frequencies (ORF's) of match stems.

The approach was found to have the following advantages:

1. *Experimental Efficiency.* Since the numeric coefficients in a regression equation are determined simultaneously in one computation, trial-and-error experimentation involving the evaluation of retrieval output to optimize parameters is largely avoidable.
2. *Computational Simplicity.* For ad hoc retrieval and routing retrieval that does not involve individual stem statistics, the computational simplicity and efficiency achieved by the model at run time are comparable to that of simple vector processing retrieval models. For routing retrieval that exploits individual stem frequencies the programming is somewhat more complicated and runs slightly slower.
3. *Effective Retrieval.* The level of retrieval effectiveness as measured by precision and recall is high relative to the simple clue-types used.
4. *Potential for Well-Calibrated Probability Estimates.* In-the-ballpark estimates of document relevance probabilities suitable for output display would appear to be within reach.

Acknowledgements

We are indebted to Ray Larson and Chris Plaunt for helpful systems advice, as well as to the several colleagues already acknowledged in our Trec-1 report. The work stations used for the experiment were supplied by the Sequoia 2000 project at the University of California, a project principally funded by the Digital Equipment Corporation. A DARPA grant supported the programming effort.

References

Cooper, W. S. (1991). Inconsistencies and Misnomers in Probabilistic IR. *Proceedings Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* Chicago, 57-62.

Cooper, W. S.; Dabney, D.; Gey, F. (1992). Probabilistic retrieval based on staged logistic regression. *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* Copenhagen, 198-210.

Cooper, W. S.; Gey, F. C.; Chen, A. (1993). Probabilistic retrieval in the TIPSTER Collections: An Application of Staged Logistic Regression. *The First Text REtrieval Conference (TREC-1).* NIST Special Publication 500-207. Washington, 73-88.

Fuhr, N.; Buckley, C. (1993). Optimizing Document Indexing and Search Term Weighting Based on Probabilistic Models. *The First Text REtrieval Conference (TREC-1).* NIST Special Publication 500-207. Washington, 89-100.

Fuhr, N.; Buckley, C. (1991). A probabilistic learning approach for document indexing. *ACM Transactions on Information Systems,* 9, 223-248.

Gey, F. C. (1993). *Probabilistic Dependence and Logistic Inference in Information Retrieval* Ph.D. Dissertation, Univ. of California, Berkeley, California.

Good, I. J. (1965). *The estimation of probabilities; An essay on modern Bayesian methods.* Cambridge, Mass.; M.I.T. Press.

Hosmer, D. W.; Lemeshow, S. (1989). *Applied Logistic Regression.* New York: Wiley.

Robertson, S. E.; Maron, M. E.; Cooper, W. S. (1982). Probability of relevance: A unification of two competing models for document retrieval. *Information Technology: Research and Development* 1, 1-21.