A tool for comparative evaluation in an interactive environment

Susan Jones, Olga Vechtomova and Stephen Robertson

Centre for Interactive Systems Research, City University, London, UK

Received 12 January 2001
Revised 1 July 2002

Abstract.

IR researchers need to collect comparative data to evaluate the worth of possible system improvements, for example new techniques for query enhancement. However it is sometimes difficult to achieve strict comparability in experiments designed for an interactive environment. We describe a method for merging results from different queries and collecting relevance feedback from users in comparative experiments. The method has been implemented in scripts supporting a Web-based front-end to a probabilistic retrieval system, using a relational database to log results. Its practical use is demonstrated in an investigation of initial query enhancement with collocate terms.

1. Introduction

Evaluation of, and comparison between, different algorithms and systems has always been a major issue for IR researchers, precisely because it is a task where success or failure is relative rather than absolute. Formal evaluation is routinely based on standard measures of recall and precision, which require relevance judgements about individual retrieved documents to be made by users seeking information. However, a distinction is sometimes made between evaluations based on pure quantitative comparisons and those with a more 'diagnostic' intent, involving detailed analyses of particular results to deduce possible reasons for success or failure.

A further distinction can be made between controlled 'laboratory-based' comparisons, such as those based on the TREC (Text Retrieval Experiment Conference) test collections, and those which involve users, either spontaneously or as volunteer experimental subjects, carrying out searches on their own behalf. In the former case, the existence of predetermined queries and expert relevance judgements makes it possible to automate the evaluation process to produce reliable comparative data. In the latter case, searches are likely to be unrepeatable events using an interactive interface, making it necessary to capture and store results and relevance judgements for evaluation purposes, as the sessions proceed.

In that context, however, it is still difficult to achieve strict comparability between alternative methods, even with a willing population of experimental subjects. If we ask people to carry out the same search under two sets of conditions, their state of knowledge, and hence searching behaviour, will be different on the two occasions. Conversely if we use two separate groups of searchers, our experimental results may be affected by small differences between them, e.g. with respect to their level of subject expertise, skill and patience, which cannot be completely controlled. We report the development of a piece of software to collect reliable comparative data in interactive experiments. It works as follows:

- the user enters a query, and a search is performed using the terms which he has entered;
- an *alternative* version of the user's query is generated, and used to perform another search;
- the results of the two searches are merged and

Correspondence to: S. Jones, Centre for Interactive Systems Research, City University, Northampton Sqare, London EC1V 0HB, UK. E-mail: sa386@soi.city.ac.uk

displayed to the user, who judges each retrieved document without knowing whether it was found by the *original* or the *alternative* search;

- if appropriate, the user may opt to produce a new *expanded* query using terms extracted from relevant documents, and the remainder of the session proceeds as in normal probabilistic retrieval;
- details of the search session are captured and logged in a relational database for later analysis, allowing the results of the alternative query to be compared with those of the original and expanded queries, and its success to be evaluated.

The above is intended to be a very general description of the method, which could in principle be used to test any new searching algorithm against one already in use, to see if it performed better. In the context of the current paper, the alternative query under investigation will be one which has been enhanced with *additional* terms before the first search is undertaken. Those additional terms are words which have shown a strong tendency to co-occur with the user's query terms in the collection as a whole – often referred to as its *collocates*. The selection criteria for collocates, and their use in the current experiment, will be fully explained in Section 5 below.

2. Query enhancement in a probabilistic retrieval system

The Okapi [1] probabilistic retrieval system developed at City University is, amongst other things, a tool for evaluation: a platform on which various retrieval algorithms and modes of presentation can be implemented and compared. As a probabilistic system, it uses relevance feedback to support the process of iterative query expansion, but the data provided for that purpose can also be logged for later analysis.

An important area of investigation for probabilistic retrieval is the effectiveness of automatic *query enhancement*, particularly at the start of a search session. Whereas end-users typically enter only two or three query terms, there is some evidence [2] that, with a suitable probabilistic model, longer initial queries work better than shorter ones, and that various forms of query expansion also help. In particular, more terms provide better leverage for discriminatory ranking between hit-list documents. Thus there is an interest in exploiting other sources of information to find and suggest useful additional query terms. Some enhancement techniques which have been or are being investigated include:

- evidence from users' earlier searches [3];
- thesaurus terms related to the initial query [4];
- statistical patterns of collocation with query terms [5];
- terms from other users for whom the same documents were relevant [6];
- 'blind expansion' with terms extracted from top-ranked documents retrieved by the initial search [7].

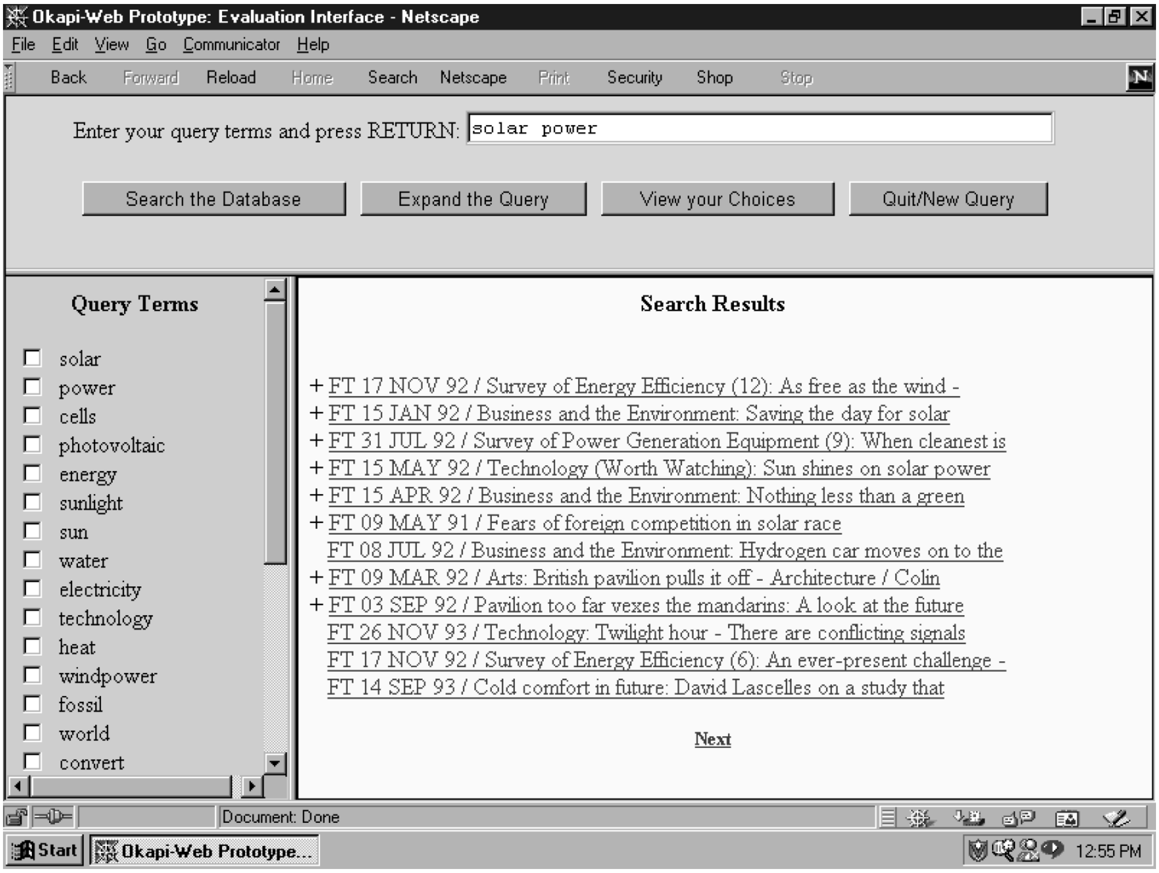
A typical evaluation procedure involves comparison of average query performance with and without the enhancement under consideration, as expressed by standard recall and precision measures. One approach is to use an existing test collection comprising a document database plus a set of queries and predefined relevance judgements. Over the last eight years the TREC test collections have been widely used for that purpose. However, it is also interesting to make the comparison in the less predictable circumstances of an interactive search session, where in practice only those items at the top of the hit-list may be seen at all, and the promotion or demotion of a few relevant documents may have a profound effect on users' perceptions. The system described in the following section is designed for that purpose.

3. Overview of the evaluation interface and query processing

The system uses a standard web browser as an interface to the Okapi retrieval system and associated functions. It presents an interface in which the screen is divided into three frames or panels – see Fig. 1. The top frame provides a space for query entry, and control buttons for selecting search functions. The narrow left frame presents a list of terms comprising the current query, and the wider right frame switches between hit-list and document display.

To maintain session continuity in the Web context, the query 'state' (terms, hit-lists, judgements) is stored in a (MySQL) relational database at the server end, and a server-side scripting language – PHP – is used to accept user input and communicate with both Okapi and the relational database. This database also acts as a repository for logging information, and supports data analysis and summary for the evaluation.

At the start of the session certain parameters are selectable, for example what kind of query enhancement is to be used. One possible choice is 'none', in which case the session reverts to non-evaluative mode. Other selectable parameters include the document


Okapi-Web Prototype: Evaluation Interface - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Search the Database
Expand the Query
View your Choices
Quit/New Query
Document: Done
Start
Okapi-Web Prototype...
12:55 PM

Enter your query terms and press RETURN: solar power

Query Terms

- □ solar
- □ power
- □ cells
- □ photovoltaic
- □ energy
- □ sunlight
- □ sun
- □ water
- □ electricity
- □ technology
- □ heat
- □ windpower
- □ fossil
- □ world
- □ convert

Search Results

+ FT 17 NOV 92 / Survey of Energy Efficiency (12): As free as the wind -
+ FT 15 JAN 92 / Business and the Environment: Saving the day for solar
+ FT 31 JUL 92 / Survey of Power Generation Equipment (9): When cleanest is
+ FT 15 MAY 92 / Technology (Worth Watching): Sun shines on solar power
+ FT 15 APR 92 / Business and the Environment: Nothing less than a green
+ FT 09 MAY 91 / Fears of foreign competition in solar race
FT 08 JUL 92 / Business and the Environment: Hydrogen car moves on to the
+ FT 09 MAR 92 / Arts: British pavilion pulls it off - Architecture / Colin
+ FT 03 SEP 92 / Pavilion too far vexes the mandarins: A look at the future
FT 26 NOV 93 / Technology: Twilight hour - There are conflicting signals
FT 17 NOV 92 / Survey of Energy Efficiency (6): An ever-present challenge -
FT 14 SEP 93 / Cold comfort in future: David Lascelles on a study that

Next

Fig. 1. Evaluation interface to the Okapi retrieval interface.

collection to be searched, the maximum size of hit-list (*maxhits*) to be presented, and the maximum number of query terms (*maxterms*) to be used in a search. User-entered query terms are stemmed, initially weighted according to their inverse frequency within the document collection, saved in the relational database, and shown in the browser in weight order. The interface supports only basic probabilistic searching: there is no provision for phrases, adjacency/boolean operators or field-based searches. The **Search** button triggers searching with both the original and alternative query, and the presentation of the merged results.

A central issue for the software design was which merging algorithm to use. Merging methods in IR have been investigated for various purposes, e.g. by participants in the TREC merging task, where the motive is to obtain the best results when different databases are searched independently. Voorhees [8], for example, discusses how performance can be optimized by taking varying proportions of records from each result set, depending on prior relevance statistics. Our own objective is different: to make an accurate comparison between two search methods rather than to produce the best result. Hence we need to present to the user an equal number of documents from the two retrieved sets, taking into account any overlap between them. The following algorithm is used:

- assume that the maximum number of hit-list documents required is *maxhits*;
- perform one search using the *original* query, taking the top *maxhits* documents, to produce set 1;
- perform a second search using the *alternative* query, again taking the top *maxhits* documents, to produce set 2;
- find the intersection between sets 1 and 2, and name the result set 3;
- remove set 3 documents from sets 1 and 2;
- keep all sets in rank order – in the case of set 3 use the higher of the two ranks;
- make up a composite hit-list by repeatedly taking one document from each of the three sets in turn; those from the intersection set come first, followed

by those from the original query and then the alternative query

- when either set 3 or sets 1 and 2 run out, continue taking documents from the remaining set(s), until there are *maxhits* documents in the hit-list, or no more documents to take.

Hit-list details are stored in the database, and the document titles presented as a set of clickable links. Following a link produces a full document display in which the original query terms, but not the additional terms, are highlighted, and a relevance judgement requested. (Users do not have the option of judging relevance purely on titles. There is evidence from earlier relevance studies, e.g. [9], that titles alone are not a good basis for relevance judgements and, since full documents are used for term extraction, it is appropriate that they should also be the basis for judgement.) The user's judgement is recorded in the database, and shown on the hit-list display in the form of a plus or minus sign next to the document title. Words (excluding stop-words) are extracted from relevant documents and stored in the database.

Once three or more relevant documents have been identified, the user may opt to create an *expanded* query using these extracted terms, as in a normal probabilistic system. When the **`Expand Query`** button is clicked, candidate terms are selected and displayed to the user, who has the option to delete unwanted terms before undertaking another search. (The threshold of three relevant documents as a basis for query expansion is a pragmatic compromise. Informal observations in earlier Okapi experiments indicated that one or two documents did not provide enough statistical evidence to produce useful results, and even with three documents we often find terms in an expanded query which are not topic-related. However, if the threshold were set higher, it would be difficult to find enough relevant documents initially to benefit from the query expansion process. No formal comparative studies have been done in this area, but since the threshold is simply another variable in the script comprising the interface software, it would be very easy to change should the need arise – it could indeed be the subject of an evaluation experiment in its own right.)

From this point on, the system operates in just the same way as it does in non-evaluative mode; terms are weighted and re-weighted according to their occurrence in relevant documents, and the composition of the expanded query may change several times in the course of a session. Meanwhile all the data about terms and documents comprising the query state and session history is logged in the relational database, so that summaries and details can be displayed at any time during or after the retrieval session.

Figure 1 shows a screen dump of the user interface. The original query (*solar power*) is shown in the top frame; terms comprising the current state of the query are shown in the left-hand frame, and a hit-list resulting from the most recent search, with a '+' next to titles which have already been judged relevant. From the point of view of a user participating in a controlled experiment, there is no indication which documents were found by which query, or that a comparative evaluation is being performed, so his or her judgement should not be influenced in any way.

4. Subjects for investigation, display of background data

The stored background data has two functions: firstly to maintain the changing query state so that the retrieval session works as it should; and secondly to support diagnostic evaluation, allowing researchers to look beyond the comparative statistics and try to discover the reasons for particular successes and failures. Following are some questions that a researcher might wish to ask about a given session:

- How much overlap was there between the original and alternative queries?
- Which query found more relevant documents?
- Where both queries found a relevant document, how did the ranks compare?
- Which terms were added to the original query by the initial enhancement?
- Which were influential in finding relevant documents?
- Which terms were added by later query expansion?
- Which were deleted by the user?

These questions motivated the design of the backend database. For a particular query session, three unique tables are generated, with the following structure:

Table: **`Terms`**
Fields: the stemmed term,
the number of postings, i.e. in how many documents it occurs
the number of relevant documents in which it has occurred so far
its weight, using the standard Robertson-Sparck Jones formula
its 'source', i.e. the unstemmed form, as entered by the user or extracted from a relevant document

its type – one of:
O: original query term
A: additional term for query enhancement
X: extracted from documents after relevance feedback
D: deleted by the user

Table: **Hits**
Fields: the document number (record key)
the document title
its type – one of:
O: retrieved by the original user's query
A: retrieved by the alternative query
I: retrieved by the original *and* the alternative query
X: retrieved by an expanded query
rank: where it appeared in the hit-list
judgement: yes/no/unjudged.

Pool is identical in format to **Hits**, but holds details of all documents judged so far during the session.

Whereas the **Hits** table is recreated in full after every search, the **Pool** table retains details of all past decisions. At the end of the session, the **Pool** and **Terms** tables are kept for later analysis. In addition, a single **Session** table is held, to accumulate summary information about *all* attempts to use the system.

Table: **Session**

Fields: a unique session identifier
the user's name
the e-mail address
the document collection being searched
the query enhancement method used
the date and time when the session started
the date and time when the session ended

The session identifier provides a link to the detailed session tables. Outside the most strictly controlled experiment, however, session records will often be incomplete, since users in a Web environment are not always willing to enter their name or e-mail address, or to end the session tidily via the **Quit** button.

The researcher can be automatically informed of the start of a session by an e-mail message specifying the session number, giving him or her the chance to look at the background data after the retrieval session, or perhaps even while it is still going on. We are aware that this facility would raise serious privacy issues in the context of general Web usage. So far, however, the system has been used only by volunteer participants who have agreed to their session being logged and observed.

The reporting script presents both summary and detail information; the examples below show how this might change as the session progresses. In this example, the *alternative* query comprises the original user's terms plus *additional* terms, The *expanded* query comprises terms extracted from relevant documents. Inevitably some of these will be terms used in the first two searches, but the re-weighting process ensures that terms are gained, lost and re-ordered following query expansion, as Tables 1 and 2 illustrate.

4.1. Summary: relevance judgements by type of query

Table 1
After an initial search and a few relevance judgements

Query type	Yes	No	Total
Original	4	2	6
Intersection	3	5	8
Alternative	2	5	7

Table 2
Following query expansion, a second search, and further relevance judgements

Query type	Yes	No	Total
Original	4	2	6
Intersection	3	5	8
Alternative	2	5	7
Expanded	8	6	14

4.2. Detail: relevant 'pool' documents

With the title of each relevant document is an indication of which type of query found it, and its rank in the hit-list in which it occurred. Since not all retrieved documents are relevant, and relevant documents are taken from several hit-lists, not every rank position is represented, and some ranks appear more than once. 'Intersection' documents (found by both the original and alternative query) show their ranks in both lists. Thus, in Table 3, a rank of 0.25 indicates that the document came top of one list but at number 25 in the other. Such documents always appear in lists according to the *higher* of their two ranks.

Table 3
Relevant documents

Type	Rank	Title
O	0	FT 14 MAY 91/Charity counts blessings amid aid fatigue/A look at
I	0.25	FT 17 NOV 92/Survey of Energy Efficiency (12): As free as the wind –
O	2	FT 09 MAY 91/Fears of foreign competition in solar race
O	3	FT 09 MAR 92/Arts: British pavilion pulls it off – Architecture/Colin
O	4	FT 21 FEB 92/Energy bill clears hurdle in Congress
X	6	FT 08 JUL 92/Business and the Environment: Hydrogen car moves on to the
I	6.41	FT 15 JAN 92/Business and the Environment: Saving the day for solar
O	7	FT 03 JAN 92/Technology: Solar power via a desert pipeline – Worth
X	9	FT 26 NOV 93/Technology: Twilight hour – There are conflicting signals
O	12	FT 15 MAY 92/Technology (Worth Watching): Sun shines on solar power
O	13	etc.

4.3. Detail: lists of terms

Term relevance counts are updated following each positive relevance judgement (Table 4). We can now see which additional terms have occurred in relevant documents, and so had a positive impact on the query, although query expansion has not yet taken place so weights have not been re-calculated. In this case the original terms have obviously been the most effective.

After three or more relevant documents have been identified, the user may opt for query expansion (Table 5). If so, the current query terms are re-weighted, and new terms extracted from relevant documents are weighted and tested for possible inclusion in the query. Terms are now ordered by their ‘selection value’ (i.e. their weight times their relevance count), and those at the top of the list will be used for the next search. In the list in Table 6, we see that most terms added to the query initially have been demoted after query expansion, replaced by extracted terms with better relevance counts. For example, SETI was an additional term used for initial query enhancement because of its association with one of the query terms using collocation measures (see Section 5.1). Because of its small number of postings, it was originally given a high weight (see Table 4), but it actually appeared in no relevant documents (see

Table 4
The initial query: *original* and *additional* terms, ranked by a weight based only on inverse frequency

Type	Postings	Weight	Term
A	8	10.1160	Seti
A	31	8.8050	Geothermal
A	32	8.7740	Biomass
A	35	8.6860	Hubble
A	40	8.5540	Ulysses
A	127	7.4070	Nasa
O	161	7.1700	Solar
A	311	6.5130	Orbit
A	808	5.5570	Powergen
A	3828	3.9870	Nuclear
A	3853	3.9800	Coal
A	6019	3.5240	Station
A	8603	3.1540	Energy
A	11 004	2.8960	Electric
A	22 110	2.1410	Parties
O	25 803	1.9660	Power
A	29 631	1.8070	Political

Table 5
Original and additional query terms with relevance counts

Type	Postings	Relevance count	Original weight	Term
O	161	13	7.1700	Solar
O	25 803	13	1.9660	Power
A	8603	8	3.1540	Energy
A	11 004	7	2.8960	Electricity
A	3853	3	3.9800	Coal
A	31	1	8.8050	Geothermal
A	32	1	8.7740	Biomass
A	3828	2	3.9870	Nuclear
A	22 110	2	2.1410	Parties
A	6019	1	3.5240	Station
A	29 631	1	1.8070	Politics
A	8	0	10.1160	Seti
A	35	0	8.6860	Hubble
A	40	0	8.5540	Ulysses
A	127	0	7.4070	Nasa
A	311	0	6.5130	Orbit
A	808	0	5.5570	Powergen

Table 6
Part of a term list after query expansion

Type	Postings	Relevance	Weight count	Selection value	Term
O	161	21	9.4510	198.471	Solar
O	25 803	20	4.1640	83.2800	Power
A	8603	15	3.4230	51.3450	Energy
X	704	8	5.4350	43.4800	Cells
A	11 004	14	2.8960	40.5440	Electricity
X	6	4	9.8130	39.2520	Photovoltaics
X	2734	9	4.0630	36.5670	Sun
X	9298	11	3.0730	33.8030	Water
X	12 075	12	2.7980	33.5760	Technology
X	159	5	6.3610	31.8050	Sunlight
X	2603	8	3.8340	30.6720	Heating
X	257	5	5.5250	27.6250	Fossil
X	7638	10	2.7320	27.3200	Efficient
X	36 751	15	1.8200	27.3000	Developments
X	2469	8	2.8230	22.5840	Panel
X	27	3	7.4070	22.2210	Non-polluting
X	48 926	12	1.7390	20.8680	World
X	4827	7	2.9040	20.3280	Fuel
X	1182	5	3.9870	19.9350	Roof
X	12 552	9	2.2100	19.8900	Environment
X	7658	8	2.4280	19.4240	Gases
X	1921	5	3.8400	19.2000	Cooled
X	48 635	13	1.4690	19.0970	Generators
X	10 851	8	2.3640	18.9120	Source
X	100	3	6.0560	18.1680	Megawatts

Table 5) – and after query expansion and term re-weighting it disappears from the current query (see Table 6).

When query expansion is activated, new terms added to the query are shown to the user, who has the option to delete any which look unhelpful. A list of deleted terms can also be seen in the background data (Table 7).

Table 7
Deleted terms

Type	Postings	Relevance count	Weight	Selection value	Term
D	43 485	17	1.8900	32.1300	Used
D	33 413	15	1.9340	29.0100	Provides
D	64 993	13	1.9930	25.9090	92
D	12	2	8.2950	16.5900	Etsu

5. An evaluation case study

A recently-completed Ph.D. thesis [10] explored the use of collocation (co-occurrence) statistics to identify words which were topically related to a set of query terms, and hence possibly useful additional terms for initial query enhancement in probabilistic retrieval. The first phase of this research, the one relevant to the current case study, is reported in Vechtomova and Robertson [5]. The experiments used the *Financial Times* 1996 portion of the TREC test collection, and the example queries were 'short titles' from TREC topics numbers 251–300. Since these were only a few words long, they could potentially benefit from the addition of extra terms.

Collocates of all query terms were extracted from 'long' text windows – typically around 100 words each – within the collection, and measures of strength of association derived for them. The most strongly associated collocates (the top eight for each query term) were added to the user's terms to create an alternative query, and the performance of the alternative query was compared with that of the original using the standard TREC evaluation measures.

In fact the enhancement method did not show any improvement on average, and other lines of investigation were subsequently followed. However, the data generated for that experiment provided ready-made test material for the interactive evaluation system described above. Moreover, using the method on an interactive system gave the opportunity to examine the process more closely, looking beyond the results of initial searches to see the relationship between initial query enhancement and normal probabilistic query expansion.

5.1. Association measures

Before presenting the results from this study, it is necessary to say a little more about the association measures used to identify potentially useful collocates of query terms. Two measures were used: Mutual Information (MI) and *Z*-score, both standard formulae being modified to take into account the fact that they counted co-occurrences within text windows averaging 100 words in length. For details and justification of the modified formulae, see Vechtomova and Robertson [5] – our immediate concern is what each formula measures and what collocational behaviour it identifies.

The MI score between a pair of words or any other linguistic units 'compares the probability that the two words are used as a joint event with the probability that they occur individually, and that their co-occurrences are simply a result of chance' [11]. The MI score increases with the frequency of word co-occurrence. If two words co-occur mainly due to chance their MI score will be close to zero.

While MI identifies associated words based on joint probability of occurrence, it gives very limited information as to how far the probability differs from chance. For that purpose the *Z*-score is a more reliable statistic, since it indicates with varying degrees of confidence whether an association is genuine, by measuring the distance in standard deviations between the expected and the observed frequency of co-occurrence. For a chance pair of low-frequency words we may misleadingly get a high MI score, whereas the *Z*-score will not be high since the variances of probabilities will be large.

The ranked lists of collocates produced by *Z*-score and MI often show very different characteristics. *Z*-score tends to identify combinations with relatively high-frequency words. The advantage of this is that they are collection-independent; the potential disadvantage is that they may emphasize syntactical structures with function words. By contrast MI highlights word combinations that are more specific, like fixed phrases and compound terms. Often these are low-frequency collection- or domain-dependent combinations, with a predominance of proper names. In the original experiments, the *Z*-score showed a slightly better performance in query enhancement overall, although as we shall see MI can produce useful individual successes.

5.2. Experimental results

TREC queries 251–300 were run through the interactive system using three different methods: once with no enhancement, once with *Z*-score enhancement and once with MI enhancement. The maximum number of query terms used in searches was set at 25, and the maximum hit-list size at 50. To save time, the official TREC relevance judgements were stored in the relational database, and an automatic judgement function was applied after the first search to identify relevant documents. If three or more such documents were found, the query was expanded, and a second search and automatic judgement was initiated. Summary data from the relational database was extracted to produce the results discussed below.

There were six queries with no officially relevant documents in the FT-96 collection, and another 14 where no relevant documents occurred within the top 50 in any of the searches. Table 8 below summarizes the results for the remaining 30 cases. The *original* query comprised terms entered by the user; the *alternative* query comprised additional collocate terms as discussed above. These two queries were used in searches to generate sets of documents; the *intersection* set comprised documents found by *both* queries. In the case where no initial enhancement took place, there is just a set for the original query. If three or more relevant documents were identified, *expanded* queries were generated using terms extracted from relevant documents, and used in a further search to generate a fourth set. Note that any document can be classified under only one of those four categories – the algorithm used makes it impossible for the same document to be counted twice. The percentage figures indicate the proportion of the 1583 TREC-relevant documents in the FT-96 collection retrieved by each search.

The three different figures for the 'original' query given here require some clarification. Where no enhancement method was used and there was no alternative query, the relevance count is based on the top 50 documents retrieved by the original query. In the other two cases, the top 50 documents were based on merged results from two searches, such that some of the relevant documents from the original search dropped below the threshold, or were counted in the 'intersec-

Table 8
Summary: number and percentage of relevant documents found by the three search methods

Enhancement		First search				Expanded search	Total
		Original	Intersection	Alternative	Subtotal		
None	Number	106			106	47	153
	Percentage	6.7			6.7	2.97	9.67
Z-score	Number	59	29	37	125	58	183
	Percentage	3.73	1.83	2.34	7.9	3.66	11.56
Mutual Information	Number	63	23	31	117	83	200
	Percentage	3.98	1.45	1.96	7.39	5.24	12.63

tion' set. Given that there must be a practical limit on the number of documents which users are willing to examine in an interactive session, the table illustrates both the gains and losses of initial query enhancement. Its effect on individual queries can be seen later in a more detailed table.

On the basis of the totals given in Table 8, searches based on both original and alternative queries appeared to perform slightly better than those using the original query alone. We may apply a simple statistical test to this summary data in order to get some idea of the validity of the observation. A chi-square (χ^2) test may be made on either recall or precision. For precision, the total number of documents retrieved over the 30 queries is 1500, and the right-hand column of Table 8 gives the number of relevant documents retrieved at the final stage by each method. Comparing *Z*-score enhancement with no enhancement, we get χ^2=2.82, with one degree of freedom, giving *p*=0.09, which cannot be taken as significant. However, comparing Mutual Information with no enhancement gives χ^2=6.79, p<0.01, which is significant. Since the total number of relevant documents is 1583, a similar analysis of recall gives almost identical results. However, the summary data conceals considerable variations between queries, as will be seen below, which makes this significance analysis a little suspect.

How much *practical* impact would initial enhancement using query term collocates have on the typical interactive retrieval session? In 11 of the 30 cases, all three methods eventually yielded the same number of relevant documents, once results from expanded searches were taken into account. Details of the remaining 19 appear in Table 9 below, which shows how many relevant documents were found at each stage of each search. It is evident that in most cases the final outcome for the three methods is still very similar, differing by only one or two relevant documents either way. Any overall improvement is largely accounted for by the four queries highlighted in the table, and in three out of those four, the improvement occurs only because the alternative query identified just enough relevant documents in the initial search to allow subsequent query expansion.

The background data for the four highlighted queries has been examined in more detail, to see which additional terms were responsible for the improved performance.

Query: World submarine forces.

> Determine the number of submarines, both nuclear-powered and conventional, presently in the inventories of all the countries in the world.

Both MI and *Z*-score alternatives performed better than the original query. Collocates associated with *submarine* (e.g. *dockyard*, *refit*, *warhead*, *devonport*, *rosyth, trident*, *polaris*, etc.) were the only effective additions, whereas those for *world* and *forces* (e.g. *Serb*, *UN, military*, *troops*, *army*, etc.) did nothing useful. For *submarine* there was considerable overlap (five out of eight) between collocates identified by *Z*-score and MI, whereas there was none at all for the other two terms. At the same time, examination of the relevant documents showed that they focused almost entirely on news stories about proposals for British submarine bases, which occurred frequently in the FT-96 collection. The alternative queries found more of these documents than the original, but did not extend it to cover other parts of the world.

Query: Environmental protection.

> Name countries that ignore or do not practice environmental protective measures.

Only the *Z*-score alternative query found enough relevant documents in the first search to allow successful expansion. The additional terms were less collection-specific than those for 'World Submarine Forces', but once again only one group of collocates had a major impact. *Environmental* brought in *pollution*, *polluted*, *emission*, *waste*, *carbon*, *waters*, *recycling* and *energy*, all of which occurred in relevant documents, whereas only *law* and *regulation*, collocates of *protection*, played any sort of a role.

Queries: For-profit hospitals, Foreign trade.

> How will the emergence of chains of for-profit hospitals affect the hospital industry and provision of health care?
>
> Define instances of the use of foreign trade as an instrument to achieve national and foreign policy objectives

MI-based enhancement was successful in both these queries, again by finding a few relevant documents initially so as to allow query expansion and a second search. In each case, however, the additional terms were extremely collection-dependent, and the success of the enhancement could be attributed almost entirely to a single acronym. One collocate for *hospital* was *HCA* (Hospital Corporation of America), which had only 25 postings but appeared in 10 relevant documents. Likewise one collocate for *trade* was *MFN* (Most Favoured Nation) which occurred in 13 relevant documents (all relating to US trade with China), and was the only added term to remain in the query after expansion.

At the detailed level, then, any successes attributable to query enhancement look somewhat fortuitous. While four or five queries out of 50 could be usefully

Table 9
Query-level detail: numbers of relevant documents found by the three search methods

	Enhancement															
Query	Relevant	None			*Z*-score						Mutual Information					
	documents	O	X	T	O	I	A	S-T	X	T	O	I	A	S-T	X	T
Exportation of industry	144	1		1	1	0	1	2		2	1	0	0	1		1
Combating alien smuggling	12	3	4	7	1	2	1	4	4	8	3	0	0	3	4	7
Environmental protection	*48*	*2*		*2*	*1*	*0*	*5*	*6*	*8*	*14*	*1*	*0*	*0*	*1*		*1*
Cigarette consumption	49	13	13	26	6	4	8	18	5	23	5	4	2	11	14	25
Algae as food supplement	6	6	0	6	1	3	0	4	1	5	3	2	0	5	1	6
US citizens in foreign jails	17	3	3	6	1	2	0	3	3	6	2	0	0	2		2
Professional scuba diving	13	1		1	0	1	0	1		1	0	1	2	3	1	4
Foreign trade	*222*	*0*		*0*	*0*	*0*	*0*	*0*		*0*	*0*	*0*	*3*	*3*	*18*	*21*
Solar power	33	19	4	23	11	3	0	14	8	22	12	3	1	16	7	23
Outpatient surgery		0		0	0	0	1	1		1	0	0	0	0		0
Ban on ivory trade	11	10	1	11	5	1	0	6	2	8	6	0	0	6	2	8
Violent juvenile crime	11	0		0	0	0	1	1		1	0	0	0	0		0
China trade	23	0		0	0	0	1	1		1	0	0	0	0		0
World submarine forces	*118*	*11*	*13*	*24*	*9*	*0*	*15*	*24*	*16*	*40*	*9*	*0*	*18*	*27*	*15*	*42*
For-profit hospitals	*20*	*2*		*2*	*2*	*0*	*0*	*2*		*2*	*0*	*2*	*3*	*5*	*10*	*15*
Foreign automobile manufacturers in US	30	0		0	1	0	0	1		1				0		0
Source of taxes	280	0		0	0	0	2	2		2	0	0	2	2		2
Worldwide welfare	42	0		0	0	0	2	2		2	0	0	0	0		0
Gun control	11	9	0	9	2	5	0	7	2	9	1	5	0	6	2	8

Note: an empty cell in this table indicates that no search with an expanded query was possible.

enhanced by one or other of the methods, an equal proportion would be likely to lose out. Some instances in the current experiment are *Algae as food supplement*, *Ban on ivory trade* and *Gun control*, where the original un-enhanced query has the better results. Here, collocates for the more general terms, e.g.

- food (retail, restaurant, drink, nestles);
- ban (whale, smoke, embargo, libya, sanction);
- trade (stock, market, export, exchange); and
- control (stake, state, shareholder, serb, system, group).

have probably been detrimental. For the majority of queries, the impact of the additional terms appears to be minimal.

6. Conclusions

Regarding the case study, the investigation confirms results produced in a non-interactive context and reported in Vechtomova [10]. Any potential improvements gained through query enhancement with additional collocate terms would not justify the extra processing required, most particularly the enormous overheads of creating and storing lists of collocates for all potential query words in a document collection. In general it exemplifies the 'swings and roundabouts' effect so often seen in attempts to improve the performance of standard probabilistic retrieval, which already exploits a large proportion of the usable statistical information (including term-dependencies) within texts.

More positively, the software described here seems to offer a useful way to perform comparative evaluations and investigate causes and effects in detail. Whether the particular interface used for these experiments is the best one for the purpose, is, of course, another question, and one which is not discussed in this paper. However, because it is created using a high-level scripting language to mediate between the web browser and the document retrieval system, it is relatively easy to adapt for different purposes. Scripts for particular functions can be dynamically selected, so in principle it should be possible to compare not only different query enhancement methods, but also alternative searching algorithms, databases or search engines. Paradoxically, however, it has so far been used for systematic evaluation only with pre-defined relevance judgements, so its potential for 'live' interactive experiment has yet to be realized. A basic mechanism for data capture has been implemented, but the logistics of its application would vary from one investigation to another, depending on the degree of control which researchers could exert over their experimental subjects. A publicly available web-based system has a potentially large number of interactive users, but any investigation methodology must allow for the fact that web users' behaviour is notoriously unpredictable, and that information about their retrieval sessions will often be incomplete. In addition, the confidentiality issue would need to be seriously addressed.

Thus the software must be seen as a set of building blocks rather than a monolithic package. To assess its viability as a practical research tool, it must be adapted for, and used in the context of, real interactive experiments. Researchers with investigations which might benefit from the approach to evaluation described here are welcome to contact the first-named author to discuss possibilities.

References

[1] S. Robertson, Overview of the Okapi projects, *Journal of Documentation* 53(1) (1997), 3–7.

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

[3] A. Goker, Context learning in Okapi, *Journal of Documentation* 53(1) (1997) 80–83.

[4] S. Jones, M. Gatford *et al.*, Thesaurus-based query enhancement. In: E.Atwell (ed.), *Knowledge at Work in Universities* (Leeds University Press, 1993), pp. 15–19.

[5] O. Vechtomova and S. Robertson, Integration of collocation statistics into the probabilistic retrieval model, *Proceedings of the 22nd BCS IRSG Meeting* (Cambridge, April 2000), pp. 165–177.

[6] M. Karamuftuoglu *et al.*, Challenges posed by Web-based document retrieval: participation of Okapi in TIPS, *Journal of Information Science* 28(1), 2002, 3–17.

[7] S. Robertson, S. Walker *et al.*, Okapi at TREC 3. In: *Overview of the Third Text Retrieval Conference (TREC-3),* NIST Special Publication 500–226 (1995), pp. 109–126. Available at: http://trec.nist.gov/pubs/trec3/t3_proceedings.html

[8] E. Vorhees, Siemens TREC-4 report: further experiments with database merging. In: *The Fourth Text Retrieval Conference (TREC-4),* NIST Special Publication 500–236 (1996), pp. 121–130. Available at: http://trec.nist.gov/pubs/trec4/t4_proceedings.html

[9] T. Saracevic, Relevance: a review of and a framework for the thinking on the notion in information science. *Journal of the American Society for Information Science* 26(6) (1975), 321–343.

[10] O. Vechtomova, Approaches to using word collocation in information retrieval. Ph.D. Thesis (City University, 2001).

[11] T. McEnery and A. Wilson, *Corpus Linguistics* (Edinburgh, 1996).