Okapi at TREC–2

S E Robertson* S Walker* S Jones* M M Hancock-Beaulieu* M Gatford*

Advisers: E Michael Keen (University of Wales, Aberystwyth), Karen Sparck Jones (Cambridge University), Peter Willett (University of Sheffield)

1 Introduction

This paper reports on City University's work on the TREC–2 project from its commencement up to November 1993. It includes many results which were obtained after the August 1993 deadline for submission of official results.

For TREC–2, as for TREC–1, City University used versions of the Okapi text retrieval system much as described in [2] (see also [3, 4]). Okapi is a simple and robust set-oriented system based on a generalised probabilistic model with facilities for relevance feedback, but also supporting a full range of deterministic Boolean and quasi-Boolean operations.

For TREC–1 [1] the "standard" Robertson–Sparck Jones weighting function was used for all runs (equation 1, see also [5]). City's performance was not outstandingly good among comparable systems, and the intention for TREC–2 was to develop and investigate a number of alternative probabilistic term-weighting functions. Other possibilities included varieties of query expansion, database models enabling paragraph retrieval and the use of phrases obtained by query parsing.

Unfortunately, a prolonged disk failure prevented realistic test runs until almost the deadline for submission of results. A full inversion of the disks 1 and 2 database was only achieved a few hours before the final automatic runs. None of the new weighting functions (Section 1.1) was properly evaluated until after the results had been submitted to NIST; we have since discovered that several of these models perform much better than the weighting functions used for the official runs, and most of the results reported herein are from these later runs.

1.1 The system

The Okapi system comprises a search engine or basic search system (BSS), a low level interface used mainly for batch runs and a user interface for the manual search experiments (Section 5), together with data conversion and inversion utilities. The hardware consisted of Sun SPARC machines with up to 40 MB of memory, and, occasionally, about 8 GB of disk storage. Several databases were used from time to time: full disks 1 and 2, AP (disk 1) and WSJ (disk 1), full disk 3. All inverted indexes included complete within-document positional information, enabling term frequency and term proximity to be used. Typical index size overhead was around 80% of the textfile size. Elapsed time for inversion of disks 1 and 2 was about two days. Running a single topic with evaluation averaged from about one minute to ten minutes, depending strongly on the number of query terms. All preliminary evaluation used the "old" SMART evaluation program. Runs tabulated in this paper used an early version of the new evaluation program, for which we are grateful to Chris Buckley of Cornell University.

2 Some new probabilistic models

Statistical approaches to information retrieval have traditionally (to over-simplify grossly) taken two forms:

(a) approaches based on formal models, where the model specifies an exact formula;

(b) ad-hoc approaches, where formulae are tried because they seem to be plausible.

Both categories have had some notable successes. A more recent variant is the regression approach of Fuhr and Cooper (see, for example, [6]), which incorporates ad-hoc choice of independent variables and functions of them with a formal model for assessing their value in retrieval, selecting from among them and assigning weights to them.

One problem with the formal model approach is that it is often very difficult to take into account the wide variety of variables that are thought or known to influence retrieval. The difficulty arises either because there is no known basis for a model containing such variables, or because any such model may simply be too complex to give a usable exact formula.

One problem with the ad-hoc approach is that there is little guidance as to how to deal with specific variables—one has to guess at a formula and try it out. This

*Centre for Interactive Systems Research, Department of Information Science, City University, Northampton Square, London EC1V 0HB, UK

problem is also apparent in the regression approach—although "trying it out" has a somewhat different sense here (the formula is tried in a regression model, rather than in a retrieval test).

The discussions of Sections 2.1 and 2.3 exemplify an approach which may offer some reconciliation of these ideas. Essentially it is to take a formal model which provides an exact but intractable formula, and use it to suggest a much simpler formula. The simpler formula can then be tried in an ad-hoc fashion, or used in turn in a regression approach. Although we have not yet taken this latter step of using regression, we believe that the present suggestion lends itself to such methods.

2.1 The basic model

The basic probabilistic model is the traditional relevance weight model [5], under which each term is given a weight as defined below, and the score (matching value) for each document is the sum of the weights of the matching terms:

$$w = \log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} \qquad (1)$$

where

N is the number of indexed documents;
n the number of documents containing the term;
R the number of known relevant documents;
r the number of relevant documents containing the term.

This approximates to inverse collection frequency (ICF) when there is no relevance information. It will be referred to below (with or without relevance information) as $w^{(1)}$.

2.2 The 2-Poisson model and term frequency

One example of these problems concerns within-document term frequency (*tf*). This variable figures in a number of ad-hoc formulae, and it seems clear that it can contribute to better retrieval performance. However, there is no obvious reason why any particular function of *tf* should be used in retrieval. There is not much in the way of formal models which include a *tf* component; one which does is the 2–Poisson model [7, 8].

The 2–Poisson model postulates that the distribution of within-document frequencies of a content-bearing term is a mixture of two Poisson distributions: one set of documents (the "elite" set for the particular term, which may be interpreted to mean those documents which can be said to be "about" the concept represented by the term) will exhibit a Poisson distribution of a certain mean, while the remainder may also contain the term but much less frequently (a smaller Poisson mean). Some earlier work in this area [8] attempted to use an exact formula derived from the model, but had limited success, probably partly because of the problem of estimating the required quantities. The approach here is to use the behaviour of the exact formula to suggest a very much simpler function of *tf* which behaves in a similar way.

The exact formula, for an additive weight in the style of $w^{(1)}$, of a term t which occurs *tf* times, is

$$w = \log \frac{(p'\lambda^{tf}e^{-\lambda} + (1-p')\mu^{tf}e^{-\mu})(q'e^{-\lambda} + (1-q')e^{-\mu})}{(q'\lambda^{tf}e^{-\lambda} + (1-q')\mu^{tf}e^{-\mu})(p'e^{-\lambda} + (1-p')e^{-\mu})} \qquad (2)$$

where

λ is the Poisson mean for *tf* in the elite set for t;
μ is the Poisson mean for *tf* in the non-elite set;
p' is the probability of a document being elite for t given that it is relevant;
q' is the probability of a document being elite given that it is non-relevant.

As a function of *tf*, this can be shown to behave as follows: it is zero for $tf = 0$; it increases monotonically with *tf*, but at an ever-decreasing rate; it approaches an asymptotic maximum as *tf* gets large. The maximum is approximately the binary independence weight that would be assigned to an infallible indicator of eliteness.

A very simple formula which exhibits similar behaviour is $tf/(tf + constant)$. This has an asymptotic limit of unity, so must be multiplied by an appropriate binary independence weight. The regular binary independence weight for the presence/absence of the term may be used for this purpose. Thus the weight becomes

$$w = \frac{tf}{(k_1 + tf)} w^{(1)} \qquad (3)$$

where k_1 is an unknown constant.

Several points may be made concerning this argument. It is not by any stretch of the imagination a strong quantitative argument; one may have many reservations about the 2–Poisson model itself, and the transformations sketched above are hardly justifiable in any formal way. However, it results in a modification of the binary independence weight which is at least plausible, and has just slightly more justification than plausibility alone.

The constant k_1 in the formula is not in any way determined by the argument. The effect of choice of constant is to determine the strength of the relationship between weight and *tf*: a large constant will make for a relation close to proportionality (where *tf* is relatively small); a small k_1 will mean that *tf* has relatively little effect on the weight (at least when $tf > 0$, i.e. when the term is present).

Our approach has been to try out various values of k_1 (around 1 may be about right for the full disks 1 and 2 database). However, in the longer term we hope to use regression methods to determine the constant. It is not, unfortunately, in a form directly susceptible to the methods of Fuhr or Cooper, but we hope to develop suitable methods.

2.3 Document length

The 2–Poisson model in effect assumes that documents (i.e. records) are all of equal length. Document length is a variable which figures in a number of weighting formulae.

We may postulate at least two reasons why documents might vary in length. Some documents may simply cover more material than others; an extreme version of this hypothesis would have a long document consisting of a number of unrelated short documents concatenated together (the "scope hypothesis"). An opposite view would have long documents like short documents, but longer: in other words, a long document covers a similar scope to a short document, but simply uses more words (the "verbosity hypothesis").

It seems likely that real document collections contain a mixture of these effects; individual long documents may be at either extreme or of some hybrid type. All the discussion below assumes the verbosity hypothesis; no progress has yet been made with models based on the scope hypothesis.

The simplest way to deal with this model is to take the formula above, but normalise *tf* for document length (dl). If we assume that the value of k_1 is appropriate to documents of average length ($avdl$), then this model can be expressed as

$$w = \frac{tf}{(\frac{k_1 \times dl}{avdl} + tf)} w^{(1)} \qquad (4)$$

A more detailed analysis of the effect on the Poisson model of the verbosity hypothesis is given in Appendix 7.4. This shows that the appropriate matching value for a document contains two components. The first component is a conventional sum of term weights, each term weight dependent on both *tf* and dl; the second is a correction factor dependent on the document length and the number of terms in the query (nq), though *not* on which terms match. A similar argument to the above for *tf* suggests the following simple formulation:

$$correction\ factor = k_2 \times nq \frac{(avdl - dl)}{(avdl + dl)} \qquad (5)$$

where k_2 is another unknown constant.

Again, k_2 is not specified by the model, and must (at present, at least) be discovered by trial and error. Values in the range 0.0–0.3 appear about right for the TREC databases (if natural logarithms are used in the term-weighting functions[1]), with the lower values being better for equation 4 termweights and the higher values for equation 3.

2.4 Query term frequency and query length

A similar approach may be taken to within-query term frequency. In this case we postulate an "elite" set of queries for a given term: the occurrence of a term in the query is taken as evidence for the eliteness of the query for that term. This would suggest a similar multiplier for the weight:

$$w = \frac{qtf}{(k_3 + qtf)} w^{(1)} \qquad (6)$$

In this case, experiments suggest a large value of k_3 to be effective—indeed the limiting case, which is equivalent to

$$w = qtf \times w^{(1)} \qquad (7)$$

appears to be the most effective.

We may combine a formula such as 6 or 7 with a document term frequency formula such as 3. In practice this seems to be a useful device, although the theory requires more work to validate it.

2.5 Adjacency

The recent success of weighting schemes involving a term-proximity component [9] has prompted consideration of including some such component in the Okapi weighting. Although this does not yet extend to a full Keen–type weighting, a method allowing for adjacency of some terms has been developed.

Weighting formulae such as $w^{(1)}$ can in principle be applied to any identifiable and searchable entity (such as, for example, a Boolean search expression). An obvious candidate for such a weight is any identifiable phrase. However, the problem lies in identifying suitable phrases. Generally such schemes have been applied only to predetermined phrases (e.g. those given in a dictionary and identified in the documents in the course of indexing). Keen's methods would suggest constructing phrases from all possible pairs (or perhaps larger sets) of query terms at search time; however, for queries of the sort of size found in TREC, that would probably generate far too many phrases.

The approach here has been to take pairs of terms which are adjacent in the query as candidate phrases.

[1] To obtain weights within a range suitable for storage as 16–bit integers, the Okapi system uses logarithms to base $2^{0.1}$

The present Okapi allows adjacency searches, so a phrase that is not specifically indexed can be searched, and assigned a weight in the usual Okapi fashion as if it had been indexed.

One problem with that approach is that the single words that make up the phrase will probably also be included in the query, and that suggests that a document which contains the phrase will be overweighted, as it will be given the weight assigned to the phrase in addition to the individual term weights. So in the present experiments the weight assigned to the phrase has been adjusted downwards, by deducting the weights of the constituent terms, to allow for the fact that the individual term weights have necessarily been added. Where this correction would give a negative weight to the phrase, it has been adjusted again to an arbitrary small positive number.

2.6 Weighting functions used

More than 20 combinations of the weighting functions discussed above were implemented at one time or another. Those mentioned in this paper are listed here. For brevity, most of the functions are referred to as **BM**nn (**B**est **M**atch).

BM0: Flat, or quorum, weighting. Each term is given the same weight.

BM1: $w^{(1)}$ termweights.

BM15: 2–Poisson termweights as equation 3 with document length correction as equation 5.

BM11: 2–Poisson termweights with document length normalisation as equation 4[2].

3 Document processing

For TREC–1 City used an elaborate 25–field structure which was intended to make all the disparate datasets on the CDs fit a unified model. It would, for example, have been possible to restrict searches to "title", "headline" etc. In the event only the TEXT was used. For TREC–2, fields which looked useful for searching were simply concatenated into one long field. For most datasets fields other than DOCNO and TEXT were ignored, but the SJM LEAD PARAGRAPH, the Ziff SUMMARY and a few additional fields from the Patents records were included. This was done using a simple *perl* script (in contrast to the TREC–1 conversion program which used *lex*, *yacc* and *C*). Most of the known data errors were handled satisfactorily, although for some reason there still remained a few duplicate DOCNOs from disk 1 and/or 2.

[2] In theory there was also an equation 5 document length *correction*, but the best value of k_2 was found to be zero.

4 Automatic query processing

4.1 Ad-hoc

A large number of evaluation runs have been done to investigate

- the effect of query term source
- the use of a query term frequency (*qtf*) component in term weighting, and
- the use of algorithmically derived term pairs.

4.1.1 Derivation of queries from the topics

Topic processing was very simple. An program (written in *awk*) was used to isolate the required topic fields, which were then parsed and the resulting terms stemmed in accordance with the indexing procedures of the database to be searched. A small additional stop list was applied to the NARRATIVE and DESCRIPTION fields only. If required, the procedure also output pairs of adjacent terms which occur in the same subfield of the topic and with no intervening punctuation. For example the command

get_qterms 70 trec12_93 tcd pairs=1

applied to

<title> Topic: Surrogate Motherhood
<desc> Description:
Document will report judicial proceedings and opinions on contracts for surrogate motherhood.
<con> Concept(s):
1. surrogate, mothers, motherhood
2. judge, lawyer, court, lawsuit, custody, hearing, opinion, finding
(topic 70)

gave

```
70:19:desc:1:contract:1
70:19:con:1:court:1
70:19:con:1:custodi:1
70:19:con:1:find:1
70:19:con:1:hear:1
70:19:con:1:judg:1
70:19:desc:1:judici:1
70:19:con:1:lawsuit:1
70:19:con:1:lawyer:1
70:19:con:1:mother:1
70:19:tit:1:motherhood:3
70:19:con:1:opinion:2
70:19:desc:1:proceed:1
70:19:tit:1:surrog:3
70:19:desc:2:contract:surrog:1
```

70:19:desc:2:judici:proceed:1
70:19:desc:2:opinion:contract:1
70:19:desc:2:proceed:opinion:1
70:19:tit:2:surrog:motherhood:2

where the fields are *topic number*, *topic length* (number of terms counting repeats but not pairs), *source field* (in precedence order TITLE > CONCEPTS > NARRATIVE > DESCRIPTION > DEFINITIONS), *number of terms*, *term* ..., *frequency* of this term or pair in the topic.

4.1.2 Document and query term weighting

Table 1 shows the effect of varying query term source fields when no account is taken of within-query term frequency.

Some tentative conclusions can be drawn: adding TITLE to CONCEPTS improves most measures slightly; TITLE alone works well in a surprising proportion of the topics; the DESCRIPTION field is fairly harmless used in conjunction with CONCEPTS, but NARRATIVE and DEFINITIONS are detrimental. (TIME and NATIONALITY fields, which are occasionally present, were never used.) This really only confirms what may be evident to a human searcher: that CONCEPTS consists of search terms, but most of the other fields apart from TITLE are instructions and guidance to relevance assessors. A sentence such as "To be relevant, a document must identify the case, state the issues which are or were being decided and report at least one ethical or legal question which arises from the case." (from the NARRATIVE field of topic 70) can only contribute noise.

However, when a within-query term frequency (*qtf*) component is used in the term weighting, the information about the relative importance of terms gained from the use of all or most of the topic fields seems to outweigh the detrimental effect of noisy terms such as "identify", "state", "issues", "question". Some results are summarised in Table 2. A number of values of k_3 were tried in equation 6, and a large value proved best overall, giving the limiting case (equation 7), in which the term weight is simply multiplied by *qtf*.

Many combinations of the weighting functions discussed in Section 1.1, as well as others not described here, were first tested on the AP and/or WSJ databases. Some of them were eliminated immediately. The function defined as BM15 gave almost uniformly better results than $w^{(1)}$, after suitable values for the constants had been found. BM11 appeared slightly less good than BM15 on the small databases, but later runs on the large databases showed that, with suitable choice of constants, it was substantially, though not uniformly, better. This may be a consequence of the greater variation in document lengths found in the large databases. Table 3 compares the more elaborate term weighting functions with the standard $w^{(1)}$ weighting and with a baseline coordination level run.

Some work was done on the addition of adjacent pairs of topic terms to the queries (see Section 2.5). A number of runs were done, using several different ways of adjusting the "natural" weights of adjacent pairs. There was little difference between them, and the results are at best only slightly better than those from single terms alone (Table 3). There was also little difference between using all adjacent pairs and using only those pairs which derive from the same sentence of the topic, with no intervening punctuation.

4.2 Routing

Potential query terms were obtained by "indexing" all the known relevant documents from disks 1 and 2; the topics themselves were not used (nor were known non-relevant documents). These terms were then given $w^{(1)}$ weights and *selection values* [11] given by $\frac{r}{R} \times w^{(1)}$ where r and R are as in equation 1.

A large number of retrospective test runs were performed on the complete disks 1 and 2 database, in which the number of terms selected and the weighting function were the independent variables. Overall, there was little difference in the average precision over the range 10–25 terms. This is consistent with the results reported by Harman in [10]. With regard to weighting functions, BM1 was slightly better than BM15. However, looking at individual queries, the optimal number of terms varied between three (several topics) and 31 (topic 89) with a median of 11; and BM15 was better than BM1 for 27 of the topics.

Two sets of official queries and results were produced. For the **cityr1** run, the top 20 terms were selected for each topic and the weighting function was BM1. For **cityr2** the test runs were sorted for each topic by precision at 30 documents within recall within average precision, and the "best" combination of number of terms and weighting function was chosen. When evaluated retrospectively against the full disks 1 and 2 database the **cityr2** queries were about 17% better on average precision and 10% better on recall than the **cityr1**. The official results (first and second rows of Table 4) show a similar difference. Later, both sets of queries were repeated using BM11 instead of the previous weighting functions (third and fourth rows of the table). These final runs both show substantially better results than either of the official runs.

Table 1: Effect of varying query term sources (no query term frequency component)

Query source	Query ave lgth	Ave Prec	Prec at 5	Prec at 30	Prec at 100	R-Prec	Recall	% of tops where AveP $\geq$ median
TC	30.3	0.300	0.624	0.536	0.440	0.349	0.683	66
C	26.7	0.296	0.636	0.524	0.436	0.346	0.686	58
TCD	39.7	0.297	0.592	0.519	0.429	0.340	0.667	62
TCND	81.0	0.263	0.612	0.485	0.394	0.306	0.605	48
TCN	71.6	0.262	0.624	0.481	0.397	0.309	0.604	50
TCNDDef	86.3	0.257	0.580	0.468	0.387	0.303	0.604	46
TN	44.9	0.181	0.500	0.418	0.320	0.245	0.491	26
TND	54.4	0.179	0.492	0.403	0.317	0.243	0.491	24
TD	13.1	0.170	0.428	0.381	0.297	0.244	0.492	28
T	3.6	0.165	0.380	0.343	0.271	0.233	0.471	32
Terms: single. Document termweights: BM11. Database: disks 1 and 2. Topics 101–150								
Query average length is the average number of terms taking account of repeats								

Table 2: Effect of varying query term sources (with query term frequency component)

Query source	Weight function	AveP	P5	P30	P100	RP	Rcl	% of tops where AveP $\geq$ median
TCND	BM11	0.360	0.652	0.569	0.479	0.401	0.754	92
TCN	BM11	0.356	0.644	0.565	0.482	0.399	0.749	92
TCNDDef	BM11	0.354	0.648	0.559	0.474	0.395	0.751	92
TCD	BM11	0.353	0.644	0.565	0.481	0.394	0.750	90
TC	BM11	0.335	0.636	0.560	0.468	0.375	0.723	86
TC	BM15	0.284	0.560	0.485	0.416	0.336	0.685	56
TND	BM11	0.283	0.556	0.503	0.414	0.338	0.652	60
TN	BM11	0.274	0.556	0.497	0.399	0.331	0.643	56
TC	BM1	0.232	0.504	0.435	0.361	0.289	0.601	28
Document term weights were multiplied by qtf, equivalent to large k_3 in eqn 6								
Terms: single. Database: disks 1 and 2. Topics 101–150								

Table 3: Effect of different document term weighting functions: single terms and adjacent pairs

Weight function	Terms	AveP	P5	P30	P100	RP	Rcl	% of tops where AveP $\geq$ median
BM11	singles + "natural" pairs	0.307	0.628	0.541	0.448	0.358	0.696	62
BM11	singles + all adj pairs	0.304	0.612	0.544	0.447	0.357	0.694	62
BM11	singles	0.300	0.624	0.536	0.440	0.349	0.683	66
BM15	singles	0.227	0.500	0.434	0.351	0.285	0.595	38
BM1	singles	0.199	0.468	0.416	0.326	0.261	0.542	22
BM0	singles	0.142	0.412	0.336	0.270	0.209	0.411	12
"Natural" means adjacent in the same sentence of the topic with no intervening punctuation								
Query term source: TC. qtf component: none. Database: disks 1 and 2. Topics: 101–150								

Table 4: Some routing results

Weight function	Number of terms	AveP	P5	P30	P100	RP	Rcl	% of tops where AveP $\geq$ median
BM1/BM15	variable	0.356	0.692	0.561	0.449	0.388	0.680	78
BM1	top 20	0.315	0.628	0.533	0.432	0.361	0.648	70
BM11	variable	0.394	0.700	0.599	0.481	0.429	0.713	92
BM11	top 20	0.362	0.684	0.605	0.459	0.397	0.707	80
Best predictive run for comparison (BM11, *qtf* with large k_3, source TCD)								
		0.300	0.612	0.524	0.394	0.345	0.632	68
Database: disk 3. Topics: 51–100								

5 Manual queries with feedback

5.1 The user interface

The interface allowed the entry of any number of *find* commands operating on "natural language" search terms. By default, the system would combine the resulting sets using the BM15 function described in Section 2.6, but any operation specified by the searcher would override this. All user-entered terms were added to a pool of terms for potential use in query expansion. Every set produced had any documents previously seen by the user removed from it.

The *show* (document display) command displayed the full text of a single document (or as much as the user wished to see) with the retrieval terms highlighted (sometimes inaccurately). Unless specified by the user this would be the highest-weighted remaining document from the most recent set. At the end of a document display the relevance question

"Is this relevant (y/n/?)"

appeared; the system counted documents eliciting the "?" response as relevant[3]. The DOCNO was then output to a results file, together with the iteration number.

Once some documents had been judged relevant the *extract* command would produce a list of terms drawn from the pool consisting of user-entered terms and terms extracted from all relevant documents. Terms in the pool were given $w^{(1)}$ weights. User-entered terms were weighted as if they had occurred in four out of five fictitious relevant documents (in addition to any real relevant documents they might have been present in). Thus for user-entered terms the numerator in equation 1 becomes $(r+4+0.5)/(R+5-r-4+0.5)$ [2].

Query expansion terms were selected from the term pool in descending order of the selection value [11] $termweight \times (r+4)/(R+5)$ for user-entered terms, otherwise $termweight \times r/R$, subject to not all documents containing the term having been displayed, and the term not being a semi-stopword[4] (unless it was entered by the user). A maximum of 20 terms was used. These selected terms were then used automatically in an expansion search, again with the BM15 weighting function.

Each invocation of *extract* used all the available relevance information, and there was no "new search" command. This was intended to encourage compliance with the TREC guidelines; it was not possible for a dissatisfied user to restart a search. When the searcher decided to finish, after some sequence of *find*, *show* and *extract* commands, the *results* command invoked a final iteration of *extract* (provided there had been at least three positive relevance judgments). Finally, the top 1000 DOCNOs from the current set were output to the results file. Apart from the aforementioned commands, users could do *info sets* and *history*.

5.2 Searchers and search procedure

The searches were done by a panel of five staff and research students from City University's Department of Information Science. Search procedure was not rigidly prescribed, although some guidelines were given. There was a short briefing session and searchers were encouraged to experiment with the system before starting. Procedures seemed to be considerably influenced by individual preferences and styles. Some searches were done collaboratively.

Searchers tried to find relevant documents by any means they liked within a single session. The number of iterations of query expansion varied between zero and four, with a mean of two. The IDs of all documents looked at were output to the results file, together with the iteration number. At the end of the session, if at least three relevant documents had been found the system did a final iteration of query expansion and output

[3] It was possible for searchers to change their minds about the relevance of a document. Subsequent feedback iterations handled this correctly, but the DOCNO would be duplicated in the search output. This appears to have led to some minor errors in the frozen ranks evaluation in a few topics.

[4] Semi-stopwords are words which, while they may be useful search terms if entered by a user, are likely to be detrimental if used in query expansion: numerals, month-names, common adverbs etc.

the top 1000 IDs; if less than three the top 1000 from the set which was finally "current" were output.

There seemed to be an impression that the new topics (topics3) are more difficult than the old. Results may also have been affected by the huge stoplist which was being used at that time because of a breakdown of the only disk large enough to hold the very large scratch files generated during inversion. Lack of the number "6" affected one topic, days of the week another ("Black Monday"). The searcher was urged to leave "Black Monday" to the end in case we were able to reindex before the deadline, but she decided to try it and thought it worked quite well.

An edited transcript of one searcher's notes is given below as Appendix B.

5.3 Results

The official results of the manual run (Table 5) are disappointing, with average precision 0.232 (60% of topics below median), precision at 100 docs 0.4 and recall 0.59. The final iteration was later re-run with BM11 instead of BM15, and the results combined with the feedback documents from the original searches for a frozen ranks evaluation[5]. This did somewhat better on a majority of the topics, but overall the manual results were very poor compared to some of the automatic runs.

6 Other experiments

6.1 Query modification without relevance information

Some iterative automatic ad hoc runs were done in which the top 10–50 documents obtained by the best existing method were used (a) as a source of additional terms and (b) as a source of "relevance" information for the $w^{(1)}$ weight calculation.

Expansion terms were selected as described in Section 4.2, in descending order of $\frac{r}{R} \times w^{(1)}$. The maximum number of additional terms was set at half the number of query terms. For many of the topics most of the top terms extracted from the feedback documents were in any case topic terms, so the number of additional terms was small.

Example (topic 112)

Topic 112: Funding biotechnology
30 feedback documents used
In the table which follows, term sources are given either as *doc*, in the case of expansion terms, or as a topic field, where *tit* > *con* > *nar* > *desc*. In this example, final weights involve a *qtf* component, and were obtained using equation 6 with $k_3 = 8$ (the resulting weight was multiplied by k_3 to obtain adequate granularity in an integer representation). For expansion terms, *qtf* was taken as 1 and the same correction applied.

				Weights		
Term	Src	*qtf*	# docs	Orig	$w^{(1)}$	Final
biotechnologi	tit	9	30	765	145	614
invest	con	4	29	148	80	213
fund	tit	2	23	78	55	88
capit	nar	2	21	78	51	81
pharmaceut	doc	(0)	15	-	73	64
ventur	nar	1	21	55	67	59
financi...	nar	2	17	64	36	57
startup...	nar	1	11	70	62	55
research	nar	1	26	35	61	54
financ	doc	(0)	15	-	54	48
partner	doc	(0)	17	-	55	48
drug	doc	(0)	18	-	53	47
investor	doc	(0)	19	-	52	46
provid	nar	3	14	66	21	45
firm	nar	1	22	36	50	44
technologi	doc	(0)	23	-	50	44
company...	doc	(0)	28	-	48	42
academ	nar	1	4	73	48	42
corpor	nar	2	9	76	26	41
monei	desc	1	18	37	43	38
stock	nar	1	20	33	43	38
industri...	doc	(0)	23	-	42	37
develop	doc	(0)	25	-	42	37
laboratori	nar	1	9	51	39	34
quantifi	nar	1	1	82	39	34
profit	nar	1	14	40	38	33
enterpr	nar	1	4	59	33	29
establish	nar	1	10	38	29	25
arena*	nar	2	0	148	15	24
data	nar	4	6	108	8	21
sale	nar	1	12	30	24	21
loss	nar	1	7	39	22	19
government...	nar	1	13	24	20	17
assist	nar	1	6	39	20	17
much	desc	1	11	28	20	17
answer	desc	1	2	52	16	14
follow	nar	1	7	26	9	8
rel*	desc	1	1	52	9	8
eg*	nar	1	0	67	8	7
question	desc	1	3	37	8	7
worldwid*	nar	2	0	126	4	6
division*	nar	1	2	41	6	5
figur*	nar	1	2	41	5	4

Here, nine of the 43 terms[6] are not from the topic. The starred terms were not used in the final search because their **selection value** $w^{(1)} \times \frac{r}{R}$ is zero (to the nearest integer). For this topic, the additional terms were beneficial and reweighting alone rather neutral.

[5] There were two topics where the searcher found no relevant documents, so for these topics the original results were inserted.

[6] The terms followed by ellipses represent synonym classes.

Terms	Wts	AveP	P5	P30	P100	RP	Rcl
All	Final	0.407	1.000	0.867	0.640	0.457	0.739
Topic	Final	0.362	1.000	0.733	0.620	0.440	0.698
Topic	Orig	0.373	0.600	0.800	0.700	0.433	0.680

Discussion

The main motive for experimenting with this type of query expansion is that it is one way of finding terms which are in some sense closely associated with the query as a whole. It does not fit particularly well with the Robertson/Sparck Jones type of probabilistic theory [5], the validity of which depends on pairwise independence of terms in both relevant and nonrelevant documents. However, it is clear, if only from the results in this paper, that mutual dependence does not necessarily lead to poor results.

There are many variables involved. In our rather limited experiments most of the initial feedback searches were done under the conditions of the first row of Table 2, that is with terms from title, concepts, narrative and description (there were a few runs using title and concepts only, but the results for most topics were not good); and weighting function BM11 with termweights given by equation 6 with large k_3 (1000). This gave nearly the best precision at 5 and 30 documents of any of our results. The number of feedback documents was constant across topics and was varied between 10 and 50. For the final search, terms were always weighted with BM11, but several values of k_3 were tried (including zero). Some runs used topic terms only and some used expansion terms as well. There was one run omitting narrative and description terms from the final search, but it was not among the very best and is not reported in the table. The number of terms in the final search was varied from 10 upwards, terms being selected as usual in descending order of $termweight \times \frac{r}{R}$. Some evaluations were done using frozen ranks, in case the initial searches tended to give better low precision, but this turned out not to be the case.

A few of the results are summarised in Table 6. They include results which appear better than the best otherwise obtained, but the difference is small, and these runs have not yet been repeated on the other topic sets. A *qtf* weight component is still needed (compare rows 2 and 14 of the table). The number of feedback documents is not critical. Speeding searching by using only the top 10 or 20 terms is detrimental.

It is interesting that results do not seem to be very greatly affected by the precision of the feedback set. Looking at the individual topics in the run represented by the top row of Table 6, 25 did better than in the feedback run, 18 did worse and the remainder about the same. Restricting to the 20 topics where the precision at 30 in the feedback set was below 0.5, the corresponding figures are 7, 10 and 3.

6.2 Stemming

A comparison was made on the AP database between the normal Okapi stemming which removes many suffixes and a "weak" stemming procedure which only conflates singular and plural forms and removes "ing" endings. For some weighting functions weak stemming increased precision by about 2% and decreased recall by about 1%, but the observed difference is unlikely to be significant.

6.3 Stoplists

Some runs were done on the AP database to investigate the effect of stoplist size. A small stoplist consisted of the 17 words

> **a, the, an, at, by, into, on, for, from, to, with, of, and, or, in, not, et**

and a large one contained 209 articles, conjunctions, prepositions, pronouns and verbs.

There was no significant difference in the results of the runs, but the index size was about 25% greater with the small stoplist.

7 Conclusions and prospects

7.1 The new probabilistic models

The most significant result is perhaps the great improvement in the automatic results brought about by the new term weighting models. In the ad-hoc runs, with no *qtf* component, BM15 is 14% better than BM1 on average precision and about 9% better on high precision and recall. The corresponding figures for BM11 are 51% and 34% (Table 3). For the routing runs, where a considerable amount of relevance information had contributed to the term weights, the improvement is less, but still very significant (Table 4). For the manual feedback searches (Table 5) there was a small improvement when they were re-run with BM11 replacing BM15 in the final iteration.

The drawback of these two models is that the theory says nothing about the estimation of the constants, or rather parameters, k_1 and k_2. It may be assumed that these depend on the database, and probably also on the nature of the queries and on the amount of relevance information available. We do not know how sensitive they are to any of these factors. Estimation cannot be done without sets of queries and relevance judgments, and even then, since the models are not linear, they do not lend themselves to estimation by logistic regression. The values we used were arrived at by long sequences of trials mainly using topics 51–100 on the disks 1 and 2 database, with the TREC–1 relevance sets.

Taking advantage of the very full topic statements to derive query term frequency weights gives another substantial improvement in the automatic ad-hoc results. Comparing the top row of Table 2 with the top row of Table 1, there is a 20% increase in average precision. The "noise" effect of the narrative and description fields is far more than outweighed by the information they give about the relative importance of terms (compare the "TCND" row of Table 1 with the top row of Table 2).

It remains to be discovered how well these new models perform in searching other types of database. Term frequency and document length components may not be very useful in searching brief records with controlled indexing, but one would expect these models to do well on abstracts. It is also rare to have query statements which are as full as the TIPSTER ones, so there are many situations in which a *qtf* component would have little or no effect.

7.2 Routing

Our results here (Table 4) were relatively good, and further improved when re-run with BM11. However, the TREC routing scenario is perhaps not particularly realistic, given the large amount of relevance information, which we made full use of as the sole source of query terms. In addition, the best of our runs depended on a long series of retrospective trials in which the number of query terms was varied. In a real-world situation one would have to cope with the early stages when there would be few documents and little relevance information (initially none at all). It would be necessary to develop a term selection and weighting procedure which was capable of progressing smoothly from a minimum of prior information up to a TREC-type situation. It may be possible to come up with a decision procedure for term selection using something similar to the selection value $w^{(1)} \times \frac{r}{R}$. Perhaps a future TREC could include some more restrictive routing emulations.

7.3 Interactive ad-hoc searching

The result of this trial was disappointing except on precision at 100 documents (Table 5), scarcely better than the official automatic ad-hoc run. On three topics it gave the best result of any of our runs, and two more were good, but the remaining 45 ranged from poor to abysmal. Little analysis has yet been done. For some topics it is clear that the search never got off the ground because the searcher was unable to find enough relevant documents to provide reliable feedback information, but the mean number found per topic was ten, which should have been enough to give reasonable results (cf Table 6, where ten feedback documents performs quite well). Currently, there are discussions towards a more realistic set of rules for interactive searching for TREC–3, and we hope to develop a better procedure and interface.

7.4 Prospects

Paragraphs

When searching full text collections one often does not want to search, or even necessarily to retrieve, complete documents. Our new probabilistic models do not apply to documents where the verbosity hypothesis does not apply (Section 2.3). Some of the TREC–2 participants searched "paragraphs" rather than documents, and this is clearly right, provided a sensible division procedure can be achieved. We made some progress towards developing a "paragraph" database model for the Okapi system, but there has not been time to implement it. Further work then needs to be done on methods of deriving the retrieval value of a document from the retrieval value of its constituent paragraphs.

Parameter estimation

Work is in progress on methods of using logistic regression or similar techniques to estimate the parameters for the new models.

Derivation and use of phrases and term proximity

A few results are reported in Table 3. They are not particularly encouraging. There is probably scope for further experiments in this area, not only on tuples of adjacent words but also on Keen-type [9] weighting of query term clusters in retrieved documents.

References

[1] D.K. Harman (Ed.), *The First Text REtrieval Conference (TREC–1)*. Gaithersburg, MD: NIST, 1993.

[2] Robertson S.E. *et al.* Okapi at TREC. In: [1] (pp.21–30).

[3] Walker, S. and Hancock-Beaulieu, M. *Okapi at City: an evaluation facility for interactive IR.* London: British Library, 1991. (British Library Research Report 6056.)

[4] Hancock-Beaulieu, M.M. and Walker, S. An evaluation of automatic query expansion in an online library catalogue. *Journal of Documentation, 48*, Dec. 1992, 406–421.

[5] Robertson, S.E. and Sparck Jones, K. Relevance weighting of search terms. *Journal of the American Society for Information Science, 27*, 1976, 129–146.

[6] Cooper, W. *et al.* Probabilistic retrieval in the TIPSTER collection: an application of staged logistic regression. In: [1] (pp.73-88).

[7] Harter, S.P. A probabilistic approach to automatic keyword indexing. *Journal of the American Society for Information Science, 26*, 197–206 and 280–289.

[8] Robertson, S.E, Van Rijsbergen, C.J. & Porter, M.F. Probabilistic models of indexing and searching. In Oddy, R.N. *et al.* (Eds.), *Information Retrieval Research* (pp.35–56). London: Butterworths, 1981.

[9] Keen, E.M. The use of term position devices in ranked output experiments. *Journal of Documentation, 47*, 1991, 1–22.

[10] Harman, D. Relevance feedback revisited. In: *SIGIR 92. Proceedings of the 15th International Conference on Research and Development in Information Retrieval* (pp.280–289). ACM Press, 1992.

[11] Robertson, S.E. On Term Selection for Query Expansion. *Journal of Documentation, 46*, 1990, 359–364.

A 2-Poisson model with document length component

Basic ideas

The basic weighting function used is that developed in [8], and may be expressed as follows:

$$w(\underline{\mathbf{x}}) = \log \frac{P(\underline{\mathbf{x}}|R)P(\underline{\mathbf{0}}|\overline{R})}{P(\underline{\mathbf{x}}|\overline{R})P(\underline{\mathbf{0}}|R)} \tag{8}$$

where

$\underline{\mathbf{x}}$ is a vector of information about the document;
$\underline{\mathbf{0}}$ is a reference vector representing a zero-weighted document;
R and $\overline{R}$ are relevance and non-relevance respectively.

For example, each component of $\underline{\mathbf{x}}$ may represent the presence/absence of a query term in the document (or, as in the case of formula 2 in the main text, its document frequency); $\underline{\mathbf{0}}$ would then be the "natural" zero vector representing all query terms absent. In this formulation, independence assumptions lead to the decomposition of w into additive components such as individual term weights.

A document length may be added as a component of $\underline{\mathbf{x}}$; however, document length does not so obviously have a "natural" zero (an actual document of zero length is a pathological case). Instead, we may use the average length of a document for reference; thus we would expect to get a formula in which the document length component disappears for a document of average length, but not for other lengths.

Suppose, then, that the average length of a document is Δ. The weighting formula becomes:

$$w(\underline{\mathbf{x}}, d) = \log \frac{P((\underline{\mathbf{x}}, d)|R)P((\underline{\mathbf{0}}, \Delta)|\overline{R})}{P((\underline{\mathbf{x}}, d)|\overline{R})P((\underline{\mathbf{0}}, \Delta)|R)}$$

where d is document length, and $\underline{\mathbf{x}}$ represents all other information about the document. This may be decomposed as follows:

$$w(\underline{\mathbf{x}}, d) = w(\underline{\mathbf{x}}, d)_1 \ + \ w(\underline{\mathbf{x}}, d)_2 \tag{9}$$

where

$$w(\underline{\mathbf{x}}, d)_1 = \log \frac{P(\underline{\mathbf{x}}|(R,d))P(\underline{\mathbf{0}}|(\overline{R},d))}{P(\underline{\mathbf{x}}|(\overline{R},d))P(\underline{\mathbf{0}}|(R,d))}$$

and

$$w(\underline{\mathbf{x}}, d)_2 = \log \frac{P((\underline{\mathbf{0}},d)|R)P((\underline{\mathbf{0}},\Delta)|\overline{R})}{P((\underline{\mathbf{0}},d)|\overline{R})P((\underline{\mathbf{0}},\Delta)|R)}$$

These two components are discussed further below.

Hypotheses

As indicated in the main text, one may imagine different reasons why documents should vary in length. The two hypotheses given there ("scope" and "verbosity" hypotheses) may be regarded as opposite poles of explanation. The arguments below are based on the Verbosity hypothesis only.

The Verbosity hypothesis would imply that document properties such as relevance and eliteness can be regarded as independent of document length; given eliteness for a term, however, the number of occurrences of that term would depend on document length. In particular, if we assume that the two Poisson parameters for a given term, λ and μ, are appropriate for documents of average length, then the number of occurrences of the term in documents of length d will be 2-Poisson with means $\lambda d/\Delta$ and $\mu d/\Delta$.

Second component

The second component of equation 9 is

$$w(\underline{\mathbf{x}}, d)_2 = \log \frac{P(\underline{\mathbf{0}}|(R,d))P(\underline{\mathbf{0}}|(\overline{R},\Delta))}{P(\underline{\mathbf{0}}|(\overline{R},d))P(\underline{\mathbf{0}}|(R,\Delta))} \ + \ \log \frac{P(d|R)P(\Delta|\overline{R})}{P(d|\overline{R})P(\Delta|R)}.$$

Under the Verbosity hypothesis, the second part of this formula is zero. Making the usual term-independence assumptions, the first part may be decomposed into a sum of components for each query term, thus:

$$w(t, d)_2 = \log \frac{(p'e^{-\lambda d/\Delta} + (1-p')e^{-\mu d/\Delta})(q'e^{-\lambda} + (1-q')e^{-\mu})}{(q'e^{-\lambda d/\Delta} + (1-q')e^{-\mu d/\Delta})(p'e^{-\lambda} + (1-p')e^{-\mu})} \tag{10}$$

where t is a query term and p', q', λ and μ are as in formula 2. Note that there is a component for each query term, whether or not the term is in the document.

For almost all normal query terms (i.e. for any terms that are not actually detrimental to the query), we can assume that $p' > q'$ and $\lambda > \mu$. In this case, formula 10 can be shown to be monotonic decreasing with d, from a maximum as $d \to 0$, through zero when $d = \Delta$, and to a minimum as $d \to \infty$. As indicated, there is one such factor for each of the nq query terms.

Once again, we can devise a very much simpler function which approximates to this behaviour; this is the justification for formula 5 in the main text.

First component

Expanding the first component of 9 on the basis of term independence assumptions, and also making the assumption that eliteness is independent of document length (on the basis of the Verbosity hypothesis), we can obtain a formula for the weight of a term t which occurs tf times. This formula is similar to equation 2 in the main text, except that λ and μ are replaced by $\lambda d/\Delta$ and $\mu d/\Delta$. The factors d/Δ in components such as λ^{tf} cancel out, leaving only the factors of the form $e^{-\lambda d/\Delta}$.

Analysis of the behaviour of this function with varying tf and d is a little complex. The simple function used for the experiments (formula 4) exhibits some of the correct properties, but not all. In particular, the maximum value obtained as $d \to 0$ should be strongly dependent on tf; formula 4 does not have this property.

B Extracts from a searcher's notes

Choice of search terms

Suitable words and phrases occurring in title, description, narrative, concept and definition fields were underlined—often this provided more than enough material to begin with. Sometimes they were supplemented by extra words, e.g. for a query on international terrorism I added "negotiate", "hostage", "hijack", "sabotage", "violence", "propaganda", as well as the names of known terrorist groups likely to fit the US bias of the exercise.

I did not look at reference books or other on-line databases, and tended to avoid very specific terms like proper names from the query descriptions, as I found they could lead the search astray. For instance, the 1986 Immigration Law was also known as the Simpson-Mazzoli Act, but the name Mazzoli also turned up in accounts of other pieces of legislation, so it was better to use a combination of "real" words about this topic.

In some queries, it was necessary to translate an abstract concept, e.g. "actual or alleged private sector economic consequences of international terrorism" into words which might actually occur in documents, e.g. "damage", "insurance claims", "bankruptcy", etc. For this purpose the use of a general (rather than domain-specific) thesaurus might be a useful adjunct to the system.

Like the other participants I was surprised at the contents of the stop-word list, e.g. "talks", "recent", "people", "new", but not "these"! However it was usually possible to find synonyms for stop-words and their absence was not seriously detrimental to any query.

Grouping of terms, use of operators

Given the complexity of the queries, it was obviously necessary to build them up from smaller units. My original intention was to identify individual facets and create sets of single words representing each, then put them together to form the whole query. [...] For example, for a query about the prevention of nuclear proliferation I had a set of "nuclear" words (reprocessing, plutonium, etc.), a set of "control" words (control, monitor, safeguards, etc.) and sets of words for countries (argentina, brazil, iraq, etc.) suspected of violating international regulations on this point. This proved a bad strategy—the large sets (whether ORed or BMed[7] together) had low weightings because of their collectively high frequencies, and the final query was very diffuse.

A more successful approach was to build several small, high-weighted sets using phrases with OP=ADJ or OP=SAMES[entence] (e.g. economic trends, gross national product, standard of living, growth rate, productivity gains), and then to BM them together, perhaps with a few extra singletons (e.g. decline, slump, recession). Because of the TREC guidelines, I didn't look at any documents for the small sets as I went along, although under normal circumstances I would have done so.

Our initial instructions were to use default best-matching if at all possible, rather than explicit operators. As already suggested, ADJ and SAMES were an absolute necessity given the length of documents to be searched, but AND and OR were generally avoided—on the occasions when I tried AND (out of desperation) it was not particularly useful. For one query where I thought it might be necessary (to restrict a search to documents about the US economy) it luckily proved superfluous because of the biased nature of the database, indeed it would have made the results worse as the US context of these documents was implied rather than stated.

Viewing results, relevance feedback

Normally I looked at about the top 5–10 records from the first full query. If 40% or more seemed relevant, the query was considered to be fairly satisfactory and I went on down the list trying to accumulate a dozen or so records for the extraction phase. As ... noted by other participants, there was a conflict between judging a record relevant because it fitted the query, and because it was likely to yield useful new terms for the next phase. On the one hand were the "newsbyte" type of documents containing one clearly relevant paragraph amidst a great deal of potential noise, and on the other the documents which were in the right area, contained all the right words, but failed the more abstract exclusion conditions of the query. I tried to judge on query relevance, but erred on the side of permissiveness for documents containing the right sort of terms.

The competition conditions discouraged a really thorough exploration of possibilities when a query was not initially successful. In one very bad case, having seen more than 20 irrelevant records and knowing that they would appear at the head of my output list, I felt that the query would show up badly in the [results] anyway and that it was not worth exploring further, as I might had there been a real question to answer.

[7] BM = "best match"; the default weighted set combination operation was BM15 (see Section 2.6)

Extracting new terms

I tried to get at least six relevant documents for the extraction phase, and usually managed a few more. As already noted, sets generated by term extraction contain only single words, so before looking at the new records I sometimes added in a few phrases to this set, either important ones from the original query or others which had occurred in relevant documents. The extracted sets of terms tended to be larger than the original query and certainly included items which a human searcher (at least one unfamiliar with this genre of literature) would not have thought of. It was amusing, for instance, to see "topdrawer" and "topnotch" (epithets for companies) extracted from documents about investment in biotechnology, and "leftist" (an invariable collocate for Sandanista) pulled out of documents about Nicaraguan peace talks. Some material for socio-linguistic analysis here!

My impression ... is that where the original document set from which terms were extracted was fairly coherent, the derived set [from query expansion] also had a high proportion of relevant documents. Not surprisingly, where I had scraped the barrel and tried several different routes to a few relevant documents, extraction produced equally miscellaneous and disappointing results.

Normally I went through two or three cycles of selection/extraction, but looking at fewer records each time. The set of extracted terms did not seem to change materially from one cycle to the next, and I would have expected the final result file reflected the query quite well even though the phrases had been lost.

Conclusion

In spite of the frustrations of this exercise, I found it a more interesting retrieval task than normal bibliographic searching, mainly because it was possible to see the full documents to gauge the success of the query, and use a broader range of natural-language skills to dream up potentially useful search terms.

Acknowledgments

We are most grateful to the British Library Research & Development Department and to DARPA/NIST for their financial support of this work. Our advisers have been unstintingly helpful. We blame the system, not our panel of searchers, for the poor results of the interactive trial. Above all, we wish to thank Donna Harman of NIST for her outstandingly efficient and courteous organisation and management of the TREC projects.

Table 5: Manual searches with feedback

Run	AveP	P5	P30	P100	RP	Rcl	% of tops where AveP $\geq$ median
Official (BM15)	0.232	0.492	0.468	0.400	0.297	0.591	40
Re-run (BM11)	0.247	0.480	0.477	0.411	0.315	0.607	48
Database: disks 1 and 2. Topics: 101–150							

Table 6: Some results from query modification

# fbk docs	Term source	k_3	Terms	AveP	P5	P30	P100	RP	Rcl	% of tops where AveP $\geq$ median
50	TCNDdoc	8	all	0.369	0.660	0.591	0.487	0.408	0.754	92
30	TCNDdoc	8	all	0.368	0.668	0.585	0.486	0.407	0.749	88
10	TCNDdoc	8	all	0.363	0.668	0.584	0.485	0.400	0.748	88
50	TCNDdoc	9	all	0.360	0.624	0.573	0.482	0.399	0.748	88
50	TCNDdoc	9	top 40	0.354	0.620	0.573	0.478	0.394	0.741	86
50	TCNDdoc	9	top 30	0.353	0.632	0.567	0.480	0.395	0.742	88
30	TCNDdoc	8	top 30	0.360	0.676	0.577	0.479	0.402	0.741	88
30	TCNDdoc	8	top 20	0.348	0.636	0.571	0.474	0.392	0.734	82
30	TCNDdoc	8	top 10	0.318	0.604	0.537	0.449	0.366	0.702	78
50	TCND	8	all	0.364	0.636	0.573	0.487	0.406	0.755	92
30	TCND	8	all	0.362	0.644	0.573	0.484	0.408	0.749	90
10	TCND	8	all	0.363	0.640	0.574	0.481	0.406	0.754	88
30	TCND	0	all	0.334	0.652	0.559	0.458	0.374	0.711	80
30	TCNDdoc	0	all	0.310	0.645	0.546	0.448	0.359	0.675	66
Initial feedback run for comparison (top row of Table 2)										
None	TCND	large	all	0.360	0.652	0.569	0.479	0.401	0.754	92
Retrospective run using all known relevant documents to reweight the topic terms										
Variable	TCND	0	all	0.371	0.708	0.600	0.497	0.408	0.758	92
Database: disks 1 and 2. Topics: 101–150										