A probabilistic model of information retrieval : development and comparative experiments

Part 1

K. Sparck Jones[†], S. Walker[‡] and S.E. Robertson[‡*]
[†]Computer Laboratory, University of Cambridge
New Museums Site, Pembroke Street, Cambridge CB2 3QG
ksj@cl.cam.ac.uk
[‡]Microsoft Research Limited
St. George House, 1 Guildhall Street, Cambridge CB2 3NH
{ser, sw}@microsoft.com
[*]also Department of Information Science, City University, London.

January 2000

Published in *Information Processing and Management*, 36, 200, 779-808.

Abstract

The paper combines a comprehensive account of a probabilistic model of retrieval with new systematic experiments on TREC Programme material. It presents the model from its foundations through its logical development to cover more aspects of retrieval data and a wider range of system functions. Each step in the argument is matched by comparative retrieval tests, to provide a single coherent account of a major line of research. The experiments demonstrate, for a large test collection, that the probabilistic model is effective and robust, and that it responds appropriately, with major improvements in performance, to key features of retrieval situations.

Part 1 covers the foundations and the model development for document collection and relevance data, along with the test apparatus. Part 2 covers the further development and elaboration of the model, with extensive testing, and briefly considers other environment conditions and tasks, model training, concluding with comparisons with other approaches and an overall assessment.

Data and results tables for **both parts** *are given in Part 1. Key results are summarised in Part 2.*

Keywords: information retrieval; retrieval theory; probabilistic model; term weighting; experiments

1 Introduction

The probabilistic approach to retrieval was first presented in Maron and Kuhns (1960). Since then it has been elaborated in different ways, tested and applied, especially in work by Maron and Cooper, by van Rijsbergen and his associates, by Croft and Turtle, by Fuhr, and by

Robertson and his colleagues at City University. As implemented in the City Okapi system it has been subjected to heavy testing in the very large evaluation programme represented by the (D)ARPA/NIST Text REtrieval Conferences (TRECs).

The literature on the probabilistic approach, even just that due to the authors mentioned, is by now extensive and as it is often also densely technical, it is hard to see the wood for the trees. There is however, by now, a well-understood core theory and well-established practical experience in exploiting this theory. Thus the probabilistic model that has been developed and applied at City has a firm grounding and demonstrated utility. This paper is intended to give a unified and accessible account of this particular model. It will show how the model treats retrieval concepts and responds to retrieval situations, and how the formal analysis on which the claim for the value of this approach to retrieval is based is supported by empirical evidence from substantial performance tests.

It should be noted that there are now several distinct versions of the probabilistic approach, in effect several different probabilistic models of information retrieval. This paper is primarily concerned with what we will for convenience label the City model, initially proposed in Robertson and Sparck Jones (1976), and subsequently developed to accommodate test findings and to meet an increasing range of retrieval circumstances and environments. Hereafter in this paper we will use "the probabilistic approach" to refer to the class of models and "the probabilistic model" to refer specifically to the City model. The presentation of the model has some historical reference, but we have organised the paper primarily to proceed logically from a simple starting point to a more complex reality, as follows.

We begin in Section 2, Foundations, with the basic elements of the general probabilistic model, providing just enough apparatus to motivate its subsequent specific interpretation. The key notions here are probability of relevance of a document to a user need, and hence of ranking documents on this basis. In Section 3, Test collections and measures, we present the data and performance measures used for the experiments associated with the development of the model in subsequent sections. We begin this development in Section 4, Data, by considering the specific types of information that are available to interpret the very abstract model introduced in Section 2. These are, naturally, facts about the occurrences of retrieval entities of various kinds: terms, documents, etc. Thus interpreting the model implies developing formulae for such purposes as term weighting. Further, as the types of data define the most basic environment variables for a retrieval system, instantiating the model operationally requires a choice of system parameters to cover these variables and perhaps further specialise generic formulae. The first part of Section 4 is in Part 1 of the paper, the rest and the remaining sections in Part 2. The material in Part 2 is treated only summarily here, and is presented in more detail in Sparck Jones, Robertson and Walker (1998), henceforth referred to as TR446 (1998). Thus Section 5, Elaborations, and Section 6, Environment, consider the richer model interpretation needed to exploit more productive resource possibilities, as in query expansion, or to respond to more challenging situations. But we focus on developments that have been fully explored and tested, leaving further discussion to TR446 (1998). In Section 7, Tasks, we briefly consider extensions of the model to other information management tasks, and in Section 8, Training, note contexts and modes of adaptation for the model.

The general pattern we will follow in the main sections 2, 4 and 5 is to present and motivate the essential aspects of the model interpretations; to consider instantiation issues and choices when implementing the model; and to summarise pertinent tests and their results. These tests are primarily experiments with a new collection drawn from the TREC materials, detailed in Section 3, explicitly designed to allow systematic comparisons on important points with a large test file. We also invoke some much older tests, and refer to other experiments done by City under the TREC Programme. Since our aim is to provide a coherent and integrated account of work done over a long period, we will focus on its major elements and refer to other publications for amplifying details, concentrating on City work. However as not only the general probabilistic model but many specific ideas are shared with others, in Section 9, Comparisons, we examine the relations between the City work and other theoretical and experimental research, considering chiefly that adopting an explicitly probabilistic approach but also considering other cases where what systems actually do is much like what City does. In our final Section 10, Assessment, we conclude by summing up on the results of the series of experiments presented in the earlier sections, ending with a final note on Open Issues.

The set of experiments covered in the paper is large, and they are referred to at many points throughout the paper. The tables giving the individual test runs, in the style described in Section 3, and that showing significance test results for the main series of comparisons we make, are therefore given in the Appendix. *This Appendix is attached to Part 1, but is also relevant to Part 2.* Selected results are repeated in Part 2, including a summary table of key results.

2 Foundations

In this section and Section 4 below, in particular, we present material that will be familiar to many (van Rijsbergen 1979). We are including it because, as mentioned earlier, our aim is to give a comprehensive account of our model and this material is needed to motivate later content. At the same time, to make the formal account accessible, we are eliminating fine detail that can be found elsewhere.

For convenience, our notation for model components and formula constituents is listed in Table 1, Notation.

2.1 Introduction

In summarising the foundations for the general probabilistic model we talk about presenting documents to the user as the output of searches. But it must be emphasised that while this may suggest the routine adhoc retrieval situation, the model is extremely general and allows for very different kinds of file items as documents, for all sorts of user needs, and for every variety of need statement, i.e. of information *request*. Equally, while a retrieval system necessarily depends on *description* – of documents and needs, the general probabilistic model is in principle compatible with many possible types of *initial* description and of contributing descriptive unit or *term*. From the model point of view, the nature of initial descriptions is part of the system environment, and the role of the model is to lead to the best derived *final* descriptions that are taken, in searching, to index documents and requests.

For the formal presentation which follows, however, we follow widespread convention and simply refer to initial document descriptions as documents D, and to initial request descriptions as queries Q. We will also say documents are *relevant* to (the needs underlying) queries. Furthermore, we will assume that descriptions are *decomposable* into smaller units or components. These may be thought of as terms, but at this stage we will still leave the precise character of terms open, so they may be simple or internally complex, as long as they are seen as identifiable units. Further, each term may simply be present or absent in the description, or may have some associated information such as frequency of occurrence in the document.

Table 1: Notation used in this paper

General	
$P(x)$	The probability of x
$P(x\|y)$	Probability of x given y
Basic variables	
D	Document or document description
t_i	A term
A_i	The ith attribute used to describe documents (e.g. term t_i)
a_i	The value of A_i for D
Q	Query or request description
L	Liked (i.e. relevant to query or need)
$\overline{L}$	Not liked
E	Elite (see section 4.5, Part 2)
Parameters	
$MS(D)$	Matching score of document, given by some query-document scoring function
$W(A_i = a_i)$	Weight associated with the value a_i of A_i
p_i	Probability of term t_i occurring in a document, given that it is liked
$\overline{p_i}$	Similar probability for document not liked
w_i	Weight associated with the presence of a term i
Data	
N	Size (number of documents in) the collection
n_i	Number of documents in which term i occurs
R	Number of relevant (liked) documents
r_i	Number of relevant documents in which term i occurs
TF_i	Frequency of term t_i in D
QTF_i	Frequency of term t_i in Q

Specific weighting funtions		Equation
UW	Unweighted terms (Quorum)	
CFW	Collection frequency weight (IDF)	6
RW	Relevance weight	8
CW	Combined weight	12 (Part 2)
CIW	Combined iterative weight	13 (Part 2)
$QACW$	Query-adjusted combined weight	14 (Part 2)
$QACIW$	Query-adjusted combined iterative weight	15 (Part 2)
OW	Offer weight	16 (Part 2)
Tuning parameters		
k_1	Effect of term frequency	9 (Part 2)
b	Effect of document length	10 (Part 2)
K	Combination of k_1, b and document length	11 (Part 2)

These components can be thought of as properties or *attributes* of the documents, in the sense of "attribute" used for structured databases. Thus the value of a term, as a document attribute, may be taken from the domain {*present, absent*} or from the domain of non-negative integers (representing the number of times the term occurs in the document). Other domains for this class of attribute, or other attributes with their own domains, may also be significant in the retrieval situation. This view of terms as attributes may be compared to the view of terms as the dimensions of a vector space in the SMART model (Salton 1975, Salton and McGill 1983), but does not imply distance or spatial relationship.

2.2 Probability of relevance

The probabilistic model seeks to ground retrieval in answering, for each document and each query, the Basic Question:

- What is the probability that *this* document is relevant to *this* query?

Strictly, "this document" has to be interpreted as "document with this content representation or description", i.e. we are asking about the probability that a document with this description is relevant to this query; for convenience we assume representations, and hence documents, are unique. The Basic Question also implies some assumptions about the nature of relevance. We do not propose to discuss these at any length here; however, they may be summarised as follows. "Query" is shorthand for an instance of information need, its initial verbalised presentation by the user as a request, and its expression as actually submitted for system searching (to which the term "query" is often restricted). Relevance is, strictly speaking, relevance to the need rather than to the query. Furthermore, relevance is assumed to be a binary attribute (a document is either relevant to a query/need or it is not), and one that can be attributed to a document without regard to any other documents in the system. These last two assumptions are very clearly oversimplifications. Finally, the attribution of relevance is normally a future event as far as the system is concerned: in other words, a strict version of the Basic Question would ask about the probability that the document *will be judged* relevant to the query/need. However, it is sufficient for our present purposes to make the simplifications and to take the Basic Question at face-value.

For similar reasons, we can limit ourselves to one query at a time; a fuller discussion, covering query sets and attributes, is given in Robertson, Maron and Cooper (1982). But for each query, we have any number of documents to consider (potentially the whole collection). We treat retrieval as a ranking process: we expect the retrieval system to rank the documents in the collection, leaving the user to examine the ranked list from the top, as far as he or she wants to go.

The idea of ranking the documents has a specific justification within the probabilistic model of retrieval (as given below). But it is also a general response to a variety of observations about the retrieval situation. For example, retrieval is inherently uncertain; some items look more similar to the query or are more promising as candidates for presentation to a user than others; some items may be more relevant than others; ranking gives the user control over how much material they have to look at; a user may want a high precision search (only a few very relevant items) or a high-recall search (anything that might be relevant) or something in between, etc (Robertson and Belkin 1978). Full ordering is not necessarily implied: a partial ordering (with tied ranks) is a form of ranking. It may be that there is not enough information for full ordering, and also that there are forms of the retrieval task for which ordering is inappropriate or insufficient but such cases should be seen as special cases or simple extensions of the general one. Some further discussion may be found in TR446 (1998, Section 7).

Under the probabilistic approach, ranking has a very specific justification and interpretation. The purpose of asking the Basic Question is to rank the documents in order of their probability of relevance. This follows from the *Probability Ranking Principle* (Robertson 1977):

P1 : If retrieved documents are ordered by decreasing probability of relevance on the data available, then the system's effectiveness is the best to be gotten for the data.

This Principle can be related to a plausible decision rule that a user might apply to decide whether or not to examine a document. Van Rijsbergen (1979) develops the rule and then relates the Principle to it. We prefer the alternative, taking the Principle as our foundational starting point and drawing an account of document description and scoring for query-document matches from it. This approach does not lead explicitly to the decision rule; but it can easily be developed to do so, though we will not fill in the detail here. The key point about the Principle, however, is that the probability of relevance is not an end in itself, but a means to rank the documents for the user. Indeed we can use any suitable transformation of the probability of relevance, rather than the probability itself, provided only that the transformation is order-preserving.

2.3 Information about documents

It is now necessary to examine what we might mean by "*this* document" in the Basic Question about probability of relevance. Every document may be assumed to be individual and unique; we can also take it that document descriptions are unique, though this naturally depends in practice on the richness of the descriptions. Whole descriptions, as unique events, do not provide much leverage for a probabilistic approach to a retrieval strategy, since it is difficult to assign probabilities to unique events. But we can exploit the decomposition of descriptions into their components or attributes. We will seek to treat individual attribute values as predictors of relevance, and to synthesize a probability of relevance for each unique document from its non-unique attribute values. Thus by "*this* document" we mean document described in this particular way, that is by this particular combination of attribute values.

2.4 Formal model

If we have some document D and query Q, we have two events:

1. L, that D is *liked*, i.e. is relevant to Q[1]
2. $\overline{L}$, that D is not *not liked*, i.e. is not relevant to Q

D might be defined as the event "we consider a document with description D", and L as the event "the user judges document D to be relevant". We would like to calculate the probability $P(L|D)$, i.e. the probability that a document is liked given that it has whatever description

[1] We use "liked" rather than "relevant" because an abbreviation R rather than L would be very inconvenient later.

it has. But in order to allow for the later expansion from D to the attributes of D, we apply Bayes' Theorem and express $P(L|D)$ in terms of $P(D|L)$:

$$P(L|D) = \frac{P(D|L)P(L)}{P(D)}$$

Moreover, since using this formula as it stands would require a further expansion of $P(D)$ beyond what we want, we simplify to avoid this by using the odds rather than the probability. Further, for reasons which will become apparent below, we use log-odds rather than odds. Log-odds can be derived from probability by an order-preserving transformation, and thus satisfy the Probability Ranking Principle given above.

$$\begin{aligned} \log \frac{P(L|D)}{P(\overline{L}|D)} &= \log \frac{P(D|L)P(L)}{P(D|\overline{L})P(\overline{L})} \\ &= \log \frac{P(D|L)}{P(D|\overline{L})} + \log \frac{P(L)}{P(\overline{L})} \end{aligned} \tag{1}$$

We now introduce the idea of matching score, *MS*, as a function of descriptions, and specifically *MS*(D) as the score for an individual document. In our presentation *MS* has a role analogous to (van Rijsbergen 1979) retrieval decision rule g. It will be convenient later to give different formulae mnemonic labels with the general form *MS-label*, so we begin this here with the most primitive case and define

$$\mathit{MS\text{-}PRIM}(D) = \log \frac{P(D|L)}{P(D|\overline{L})}$$

MS-PRIM is a function of the whole document description D; we plan to expand it later into a function of the attributes of D. By equation 1,

$$\mathit{MS\text{-}PRIM}(D) = \log \frac{P(L|D)}{P(\overline{L}|D)} - \log \frac{P(L)}{P(\overline{L})}$$

Since the last term is the same for all documents, a ranking of documents in *MS-PRIM* order is a ranking in $P(L|D)$ order. Thus given an estimate of *MS-PRIM* for each document, we can use it to rank documents in the proper order. (We will however be making one further order-preserving transformation before finalising on the basic document scoring formula.)

2.5 Independent attributes

The way the general model has normally been developed has been by making the very strong *Independence Assumption*, I1, about the the attributes defining the system's universe of documents:

I1 : Given relevance (likedness), the attributes are statistically independent.

That is, within each class of documents (defined by relevance, i.e. L or $\overline{L}$), each attribute is statistically independent of all the other attributes

This assumption is patently not justified in reality, especially in the fine grain (van Rijsbergen 1979)see e.g.. However it has three important merits. First, it makes the formal development and expression of the model easier; second, it makes model instantiation and system operation tractable; and third, it still permits indexing and searching strategies that improve performance compared with the rudimentary baseline strategy, namely simple term matching, that does not exploit the model. It has also been shown that the model can be developed with a somewhat weaker set of assumptions (Cooper 1995). We discuss the model without the Independence Assumption later.

Under the Assumption, we have a very simple derivation of document probability from attribute probabilities, with

$$\begin{aligned} P(D|L) &= \prod_i P(A_i = a_i|L) \\ P(D|\overline{L}) &= \prod_i P(A_i = a_i|\overline{L}) \end{aligned}$$

Here A_i is the ith attribute, and a_i is its value for the specific document. The product is taken over a set of appropriate attributes: we discuss the way these are defined later. Now, we can recast *MS-PRIM* as

$$\textit{MS-PRIM}(D) = \sum_i \log \frac{P(A_i = a_i|L)}{P(A_i = a_i|\overline{L})}$$

This equation implies that (under the Independence Assumption) we could calculate a score for each document, made up as a sum of parts, one relating to each attribute of the description. This looks very convenient; however, we can make it more convenient still by ensuring that "natural" zero values can be taken as zero. The formula as it stands requires an explicit component to be included for every value of an attribute, for example for the absence of a term as well as for its presence. It would be more straightforward to include values for term presence only, regarding term absence as a natural zero. This can be achieved as follows.

For every attribute which has such a natural zero, we subtract the component relating to this zero value of this attribute from the score of *every* document. (Since the same quantity is being taken from every document's score, the transformation is order-preserving.) So we define a new matching score, which we will call *MS-BASIC*, where:

$$\textit{MS-BASIC}(D) = \textit{MS-PRIM}(D) - \sum_i \log \frac{P(A_i = 0|L)}{P(A_i = 0|\overline{L})}$$

Then

$$\begin{aligned} \textit{MS-BASIC}(D) &= \sum_i \left(\log \frac{P(A_i = a_i|L)}{P(A_i = a_i|\overline{L})} - \log \frac{P(A_i = 0|L)}{P(A_i = 0|\overline{L})} \right) \\ &= \sum_i \log \frac{P(A_i = a_i|L)P(A_i = 0|\overline{L})}{P(A_i = a_i|\overline{L})P(A_i = 0|L)} \qquad (2) \end{aligned}$$

Now if we define:

$$\begin{aligned} W(A_i = a_i) &= \log \frac{P(A_i = a_i|L)}{P(A_i = a_i|\overline{L})} - \log \frac{P(A_i = 0|L)}{P(A_i = 0|\overline{L})} \\ &= \log \frac{P(A_i = a_i|L)P(A_i = 0|\overline{L})}{P(A_i = a_i|\overline{L})P(A_i = 0|L)} \qquad (3) \end{aligned}$$

it follows from equation 2 that

$$MS\text{-}BASIC = \sum_i W(A_i = a_i) \tag{4}$$

The W function now provides a *weight* for each value of each attribute, and the matching score for a document is simply the sum of the appropriate weights. $W(A_i = 0)$ is always zero, so zero values of attributes can safely be ignored. Furthermore, attributes which we have no reason to associate with relevance can also be safely ignored. For example, for a randomly chosen term, with no known relation to the query, we can reasonably assume that all weights are zero.

2.6 Term presence and absence

We can exemplify the above formal model (with the Independence Assumption) very simply, using the case where attribute A_i is simply the presence or absence of a term t_i. We will denote $P(t_i \text{ present}|L)$ by p_i and $P(t_i \text{ present}|\overline{L})$ by $\overline{p_i}$; the corresponding absence probabilities are calculated by subtracting the presence probabilities from one. Then the formula for W (equation 3) gives a weight for term presence:

$$w_i = \log \frac{p_i(1 - \overline{p_i})}{\overline{p_i}(1 - p_i)} \tag{5}$$

The matching score for the document is in turn just the sum of the weights of the matching (i.e. present) terms. This version of the weighting formula will be used extensively in what follows. Where there is no danger of confusion, the suffix i will be dropped.

3 Test collections and measures

3.1 Data sets

As we progress in the next and following sections through successive interpretations of the general model we will, as mentioned earlier, illustrate the effects of their application on retrieval performance. One function of this paper is to draw together important performance data scattered over many papers or buried in reports. But we have also exploited the accumulation of materials in the major TREC evaluations of the last decade (TREC 1992–1999) to form a new large test collection and carry out completely new experiments. The full set of results reported here thus further extends our tests beyond those reported for individual TREC cycles in e.g. City University papers in TREC, which were themselves significant advances, with respect to collection scale, on older ones applying the model.

Along with our main new TREC collection, described below, we reproduce some older results using the Cranfield, UKCIS and NPL collections. This is partly to maintain continuity with our own earlier research and to allow references to comparable work done by others, e.g. to SMART work reported in Salton and Buckley (1988, 1990); and partly to show performance comparisons across a range of environments. The historic Cranfield collection has short initial manual index descriptions based on the whole document, while the NPL collection has short automatic descriptions from abstracts. Both of these have straightforward requests in the form of natural language sentences or phrases. The UKCIS collection, in contrast, has only titles for documents, but has requests originally constructed as boolean profiles for routing, with many terms. The TREC collection has automatic initial descriptions in natural text form, mainly for the full documents but in some cases for abstracts. The TREC requests were also designed for routing, with 'title' and careful 'description' and 'narrative' fields, the description defining the topic and the narrative the conditions for relevance. The Cranfield collection is very small; UKCIS and NPL are quite large by historical standards, but the TREC collections, representing current data size norms, dwarfs them. Comparisons across these collections are needed, in the usual way, to check for some environment variation. But we also need them for a more important reason. The comparison between the three older collections on the one hand and TREC on the other is critical, for our model, in showing how performance is affected first by the shift to full text documents, and second by scaling up to much larger files with hundreds rather than tens of thousands of items. At the same time, our main experiments with the new TREC collection cover a much larger range of comparisons bearing on our concerns here than the older collections do, where we reproduce results originally published in e.g. Sparck Jones and Webster (1980).

The details given in Table 2 summarise the salient facts about the collections. The older collections are named as in the earlier literature, for the particular versions used: C1400I for Cranfield, U27000Pb for UKCIS, and N11500A for NPL. We have named the new TREC collection T741000X. We may, however, simply refer to 'the' or 'our' Cranfield, UKCIS, NPL or TREC collections where there is no ambiguity. Further information about the collection characteristics and construction (e.g. the provision of relevance assessments) is given in the references with the Table. It should be noted here, however, that the relevance data for the UKCIS collection is limited and biased towards the profiles so the level of performance, even though only titles are being searched, may appear higher than it should. There are some small variations in the basic facts (e.g. precise number of documents) given for the underlying material or specific versions of it in different publications, but these are simply due to minor administrative differences, cleanups, etc and do not affect the test results. We should also note that where we use the manually indexed version of Cranfield, SMART has used automatically processed abstracts; but this is not important in the present context. For all our tests we take stemming (Porter 1980)following, along with straightforward stop word removal, as defining *basic single term indexing*.

To form our main TREC collection we combined batches of requests from separate TREC evaluations. Thus where successive TRECs each used 50 requests ('topics') we have formed a new set of 150 requests (nos 51–200). This is the largest union set with reasonably similar construction, form and quality characteristics for which there are also relevance assessments for a large document set (these criteria exclude the later TREC requests). It has the further advantage that, since the requests in their initial state are elaborate and extensive, with several components, we have been able to compare performance for them in several *forms*, namely Long (L), Medium (M), and Very short (V). The Long forms cover all of the title, description and narrative fields, the Medium forms cover the title plus description field, and the Very short forms just the title field. [2] The Long forms resemble the UKCIS profiles in careful content; the Medium ones are nearer the natural requests used for NPL and UKCIS, but are more carefully formulated. We used the Very short requests, intended primarily as summary headings, as the nearest we could get, for our TREC collection, to the kind of brief and not necessarily carefully formulated requests often found in practice. As Sparck

[2] We use the term "Medium" rather than "Short" since the latter has been used in TREC to refer to requests consisting of descriptions only, found not to be properly autonomous.

Table 2: Test collection data

```
OLD COLLECTIONS : for further details see Sparck Jones and Webster (1980)

   Cranfield 'C1400I'
     1400 documents in aeronautics with manual word indexing
     225 requests, simple sentences or phrases
     exhaustive relevance assessments

   UKCIS 'U27000Pb'
     27361 documents in chemistry represented by titles
     75 requests, terms from elaborate SDI profiles
     relevance assessments on original profile output

   NPL 'N11500A'
     11429 documents in electronics represented by titles and abstracts
     93 requests, simple sentences or phrases
     relevance assessments from original study pooled outputs


NEW COLLECTION : for further details see Harman (1993-9)

   TREC 'T741000X'
     741856 documents in news, computing, official publications and energy
       represented by full text (over 2/3) or abstracts;
       these documents are the combined TREC Disc1 and Disc2 sets
     150 requests, words from structured profiles with sections
       title, description, narrative
       'L' long requests = title+description+narrative
       'M' medium requests = title+description
       'V' very short requests = titles only
       these requests are TREC topics 51-200
     relevance assessments from TREC evaluation pooled outputs

Collections divided into training and test halves by even/odd document
numbers. Test Half, H, is odd-numbered for old collections, even for new.

Collection statistics referring to terms are after stopping and stemming
using straightforward stop lists and Porter or Porter-style stemming.
```

Table 2 (contd): Test collection data

SUMMARY STATISTICS

	no docs	no terms	av terms/ doc	no reqs	av terms/ req	no reldocs	av reldocs/ req	rels as % of total docs
C1400I	1400	2683	29.9	225	7.9	1614	7.2	0.51%
H	700					780	3.5	
U27000P	27361	n/a	n/a	75	18.3	3739	49.9	0.18%
H	13681					1902	25.4	
N11500A	11429	7491	20.0	93	7.2	2083	22.4	0.19%
H	5715					1061	11.4	
T741000X	741856	1290709*	129.9	150	L 32.6 M 10.3 V 4.0	37819	252.1	0.03%
H	370928	839463				18927	126.1	

Ranges	C1400I	U27000P	N11500A	T741000X		
max terms/doc	102	n/a	105	14083		
av	29.9		20.0	129.9		
min	5		1	1		
				L	M	V
max terms/req	17	86	14	85	24	11
av	7.9	18.3	7.2	32.6	10.3	4.0
min	3	1	2	8	2	1
max reldocs/req	40	554	84	1141		
av	7.2	49.9	22.4	252.1		
min	1	1	1	14		

* The number of terms in T741000X is very high but there are many miscellaneous nonwords. 143778 terms beginning with a letter occur in at least 5 documents.

Jones (1999b, 1999c) illustrate, TREC test performance as a whole has declined with more 'realistic' requests in later cycles: we consider later the challenges presented by the very minimal requests often submitted in operational situations. The document file for our TREC collection combines the so-called Disk 1 and Disk 2 sets, as in the largest size file used for any TREC cycle test of retrospective searching in the main Adhoc evaluation test (the TREC–6 Very Large Collection track used about 7.4M documents but had limited relevance assessments). This document set is made up of several distinct subfiles with different subject and genre characteristics, and is therefore much more heterogeneous than those for the other test collections.

Our TREC collection, 'T741000X', is thus a larger test collection, from the request point of view, than any used hitherto in mainstream TREC, itself the largest systematic retrieval evaluation effort so far. It is also not merely very much larger than older experimental collections: it is a more substantial data set, in important respects, than that used in Lancaster's historic investigation (Lancaster 1969): Lancaster's study was impressively large in terms of document and request numbers (approx 300 x 800,000); but it did not involve either text searching, only matching on short controlled term lists, or multi-strategy comparisons.

However it is also the case that much of the work to be described addresses term weighting based on relevance feedback. Experiments on relevance feedback are a little tricky: it would not in general be an appropriate experimental strategy to conduct a search, identify relevant documents in the output, improve the query through relevance feedback, and then evaluate the improved query on the same collection. (The experiments below described as 'retrospective' are indeed of this type, but they have limited validity except as yardsticks.)

There are various experimental approaches to this problem of evaluating relevance feedback, including 'residual ranking' and half-collection experiments. We have adopted the latter: the collection is divided pseudo-randomly into two halves (e.g. even- and odd- numbered documents), relevance feedback is obtained from one half and the improved queries evaluated on the other half. While this is not a very realistic procedure, it allows for easy interpretation and for unbiased evaluation of the possible improvements to be obtained from relevance feedback. Residual ranking would be somewhat more realistic, but would in general provide less rich information about performance.

Our adoption of the half-collection paradigm naturally implies that all runs are actually on the smaller Half (H) document sets shown in Table 2 than on the full collections. Thus for the T741000X collection we have 370928 rather than all 741856 documents for the 150 requests. But even this reduced collection is respectably large as an experimental one.

The collections described, and the TREC collection in particular, are those we refer to under our main topics, and therefore illustrate one-off or 'adhoc' searches. There are, however, topics we discuss for which this data cannot be used, for instance (obviously) retrieval for languages other than English, or tasks like filtering, where both documents and relevance assessments lack the necessary temporal properties. We may therefore refer under these headings to experiments with Okapi as reported in City papers in TREC (1992–1999), as providing ancillary but not strictly comparable performance data.

In considering the effects of strategies on performance, in the next sections, we concentrate on generalisations about relative strategy performance across the different collections and, for TREC, request forms. We comment only on particular collection and collection properties where this is essential. In the later Assessment section, 10 (Part 2), we examine the impact of collection characteristics, e.g. brief documents or very short requests, on strategy performance. Further, since the number of possible strategy comparisons is very large, we concentrate in the next sections on the most pertinent ones for the local context, extending the comparisons over a wider range and drawing broader conclusions in the Assessment section.

3.2 Performance measures

The test runs given in this paper have been chosen primarily to illustrate the main points about our model, in a clear and consistent way. Our aim is to offer an overview of what the model delivers when it is applied and hence to indicate its demonstrated performance value. We do not go through all the demonstration in detail. The results shown are therefore selective in two ways. First, we have taken runs from the past just for the most straightforward and simple values for environment variables, e.g. without separating highly from partially relevant documents, and with well-attested, general-purpose instantiations for the model without tailoring for specific collection conditions. In some cases, therefore, the runs drawn from older research are not necessarily those delivering the absolutely best performance. Second, for the new work, the results are selective in the sense that we have done runs only to fill in the test grid so cross-collection comparisons can be made within the same straightforward framework, without seeking best parameter choices. The model instantiation for the TREC collection has required some parameter setting. Trials showed that it was not possible to find good values applicable across all request versions (thus reducing test variation); but the same values were used for L and M, slightly different for V. So while there has been some collection tuning, this has been of a modest kind essentially reflecting limited sampling.

Thus while the test runs given here are a selected few, the selection is the reverse of pernicious. By giving the tests as a single series we can show whether our model is robust and reliable, able to deliver respectable performance in very different environments and under some realistic conditions as far as file and request properties are concerned. Our initial *baseline* performance is that given by simple term coordination, which can be viewed as applying the simplest plausible retrieval model. Then in developing the model we address internal performance improvement and comparisons.

For the same reasons as motivate our choice of runs, we have confined ourselves here to a limited set of performance measures. We follow convention, and maintain the connection between older and newer results by referring to Precision at standard Recall levels. But we do this only in restricted fashion. We believe that these widely used figures are both opaque and misleading: the former because they obscure the actual numbers of documents, perhaps thousands, needed to get beyond low recall; the latter because assuming the entire collection is ranked may be inappropriate if negative matching scores imply the user should *not* be given output, or if there is a large mass of undifferentiated documents at some natural 0 score. Thus we give only Precision at Recall = 30%, which we may abbreviate to 'Rec30'. For the TREC runs this is drawn from levels computed in the normal SMART/TREC style (Harman 1997); for the older collections the computation followed then-SMART precedent, so in the fine-grain the procedure used, e.g. in interpolation, may not be absolutely consistent throughout (or indeed identical with that applied in other cited tests). We also give Average Precision, 'AveP', as a widely invoked, though limited, global measure; this is computed in the normal TREC manner for our main collection, but by crude approximation by averaging over the recall levels for the older ones (or in some cases is guestimated using the range of performance data actually available). [3]

We prefer, however, to focus on what we believe are more meaningful performance indicators, and have therefore used Precision at Document Cutoffs = 5,10,15,20,30,100, in the style established for the TREC Programme evaluations. This measure shows retrieval performance in a more directly comprehensible way than the recall levels one, and is also easily related to the 'pages-worth' output of Web search engines. More particularly, we select Precision at Document Cutoff 30, abbreviated as 'Doc30', as a key value for discussion purposes: this is also used for the range of TREC performance comparisons across participating teams summarised in Sparck Jones (1999b, 1999c). We give Recall at rank 1000, and also Precision at the query rank where the number of retrieved documents is the same as the number of relevant to retrieve (RPrec), primarily as information about the collection. We use the document cutoff measure for all of our main experiments and comparisons with the T741000X collection, and we also give it, for a smaller set of values, for the NPL collection, though we unfortunately cannot give it for all the older collections.

In relation to the older collections, the performance figures given here are drawn from a very much larger set to be found in the various cited references. Our aim in using them has been to focus on key comparisons, across collections, in as simple and straightforward a way as possible and taking a broad view of what the results as a whole show about performance for our model. From this point of view small historical discontinuities, or crudities like the use of truncation rather than rounding in older figures, are not an issue: we believe that the figures for our measures have enough common grounding to support the required robust general conclusions. The much larger set of TREC-collection based results can, on the other hand, be related to the many other TREC evaluation results based on the model reported in the City papers in TREC (1992–1999), and through these to work done elsewhere, as discussed in Section 9, Part 2.

We have applied obvious statistical significance tests to the TREC results, but with some reservations about their real propriety and value; and they are in any case relatively weak. Thus we have used the t test, Wilcoxon, and sign test, at 2.5% and 1% levels. The latter two non-parametric tests involve query-based pairwise comparisons: that is, for each pair of runs being compared, the two results for each query are compared first, and the comparisons are accumulated over queries. The t test is also used in a query-based pairwise fashion, that is on the distribution of per-query differences in parameter values. As it is not completely clear that the strongest of these three, the t test, is justified for the retrieval case, we have concentrated on Wilcoxon, with supporting evidence from the sign test. Thus we consider the Wilcoxon test results for all of our particular comparisons, referring to the weaker sign test only when Wilcoxon does not show a statistically significant difference. We have applied the tests to our Doc30 and Rec30 figures, and also to AveP, the latter primarily for compatibility with others. As the default we accept differences that are significant at the 2.5% level, though it should be noted that many of our differences are significant at the 1% level. Further, we are primarily interested in whether performance differences we informally characterise as at least noticeable are also statistically significant: we are not especially interested in the possibility that differences we do not see as even noticeable are nevertheless statistically so.

The details of the comparisons to which we have applied significance tests are given in Table 8 in the Appendix. For the Wilcoxon test results shown there the numerical values 2.33 and 1.96 correspond respectively to 1% and 2.5% significance levels on a one-tail test.

[3] Ranking was computationally expensive for large collections in the seventies.

To avoid overloading the text, the simple statement 'this difference is (or these differences are) statistically significant' should be read as meaning significant at the 2.5% level. We elaborate only where this is particularly appropriate. Thus it should also be noted that in these statements we cover all three request versions, and AveP as well as Doc30 and Rec30.

The significance tests apply only to the TREC results. For the older, smaller collections the data for significance testing is no longer available: this implies that for these collections even apparently large performance differences have to be treated with caution.

At the same time, since we are especially concerned in this paper with strategy comparisons that hold across a range of collection conditions, it is impossible to avoid informality in summary comments. We will therefore make some use, to encourage consistency, of earlier terminology for degrees of performance difference, namely whether this is *Noticeable* (A > B), *Material* (A >> B), *Striking* (A >>> B), or *Dramatic* (A >>>> B), which we apply here to precision differences at Rec30 or Doc30 of at least 2,4,6,8 .., full points. Thus if we say that strategy A is Materially better than strategy B, this implies that there is at least 4 points Precision difference for all the collections in question. We also may refer, even more informally, to "modest", "good" etc performance, and in seeking to characterise performance across a range of situations for different collections or request forms may refer to "large" performance differences as ones which are at least Noticeable and typically greater than this. However we support these informal characterisations with notes of significance test values. Note that in line with our emphasis on broad, solid performance differences, we simply truncate run performance values to two figures.

Our starting point for retrieval performance is therefore the baseline unweighted *UW* performance figures given in the Appendix, in Table 6 for the TREC data and Table 7 for the old collections. These show, very clearly, how low absolute performance for such a naive approach is in the TREC full text case, regardless of the differences across the request forms and whether measured by Rec30 or Doc30: given the large number of relevant documents typically to be found, 4 relevant documents on average by rank 30 is uninspired. The contrast, for Rec30, with the older collections is rather marked: the performance levels are higher for these, but can be attributed to the more favourable properties of the data, whether small collection (Cranfield), concentrated searching on abstracts (NPL), or exceptionally elaborate requests to compensate for searching titles (UKCIS). It is also noticeable (in TREC) that the longer the query, the worse the performance.

4 Data

Interpreting the general probabilistic model outlined in the previous section means using the specific kinds of distributional information that are available for terms and documents. It is also necessary to be explicit about the status of the search query, which was not directly mentioned in the last section. Thus we referred to some unspecified set of "appropriate" vocabulary terms present in a document as the basis for estimating relevance to the user, and hence deciding to retrieve, without explicitly considering their relation to the terms present in the query, even though the query is taken to represent the user's need. But while it is not necessary to assume that a query is a wholly adequate representation of a user's need, it is both natural and reasonable to take the current query as given and to concentrate specifically on the presence of query terms in documents. Query terms are the proper starting points for estimating relevance, so we should begin by considering the evidence their presence (or absence) supplies.

But before doing this it is worth noting that focussing on the query terms is not merely rational in the obvious sense just indicated. It is a larger point drawing out implications of the notion of 'this document' introduced in Section 2.3. Thus one implication is that as in retrieval documents are what users make of them, document indexing should not be presumptive of the user. This is familiar as the postcoordinate philosophy that underpins modern retrieval systems, where document descriptions are initially open and are then closed as and how they match each query. However in early automated systems, postcoordination was still associated with initial document characterisation by assigned or selected keywords or descriptors: thus a significant document indexing step occurred at file time. The general movement since to the less presumptive and more open initial descriptions represented by documents' actual words has constituted a further shift towards search time indexing as a logical and not just practical matter. In general, especially with large files, it is useful to delay work on a document until it is needed (e.g. it is worth waiting to see whether a text has something in common with a query before parsing it). But much more importantly as a second implication, in search time indexing a document's description is influenced by (even if it does not wholly depend on) the state of the file at that moment, i.e. index variables have different values at different times. This is a key difference between conventional Boolean and modern weighted searching. Search time indexing thus means more than just the use of postcoordination, and when it refers explicitly to the state of the file is dealing with the idea of evidence for document relevance which is central to the whole probabilistic model.

We can now consider what information about documents (directly or indirectly bearing on queries) is available to interpret the general model. We will continue for the present to assume unit terms as description elements, and as a concrete example take these to be single word stems of the sort that have been established as generally useful and are widely used. However the model still leaves open the methods by which these have been produced to form initial descriptions: they could be manually assigned or automatically extracted, and could be based on document surrogates (like abstracts) or on entire full texts.

4.1 Term incidence

Clearly the first and most obvious data are simply the facts about term presence, i.e. *incidence* in documents. We thus have to determine the contribution that the presence of a term in some specific document makes to that document's probability of relevance from the term's overall incidence. That is, the term's contribution will depend on the relation between the number of documents in which it occurs and the number of documents in the file. Further, the fact that the number of relevant documents for a query is normally low by comparison with file size suggests that the presence of a less common query term in a document may be a better predictor of relevance than that of a more common one. In these circumstances, a plausible weighting function for query terms is

$$CFW = \log \frac{N}{n_i} \tag{6}$$

where N is the size (number of documents in) the collection
and n_i is the number of documents containing query term i

and the matching score (equation 4) becomes

$$MS\text{-}CFW = \sum_i \log \frac{N}{n_i},$$

summed over query terms.

This weight is the familiar collection frequency weight (CFW)[4] introduced in Sparck Jones (1971) It was then justified on the basis only of the implications of incidence frequency just mentioned, without any reference to the probabilistic model, and subsequently on the basis of substantial experimental evidence. In fact, the formula (or something very similar) can be derived from equation 5 through explicit assumptions about p_i and $\overline{p_i}$, as will be seen below.

Instantiating the model for such a simple form of weight presents no problems and practical implementation is quite straightforward. Table 3[5] shows the results of applying these CFWs, using Rec30 for all the Half test collections and Doc30 as well for Half T741000X. As others have also found, CFWs can generally be expected to give a modest (statistically significant) performance improvement over the baseline. Thus while there is an exception in the Cranfield case, and the older figures are only informal estimates, for our TREC collection for all the request forms it is at least the case that $CFW > UW$ and the gain is often greater. Table 8 shows that the difference is also statistically significant. However these figures also illustrate the point that quite large percentage improvements e.g. doubling the number of relevant retrieved at rank 30 from 1 to 2, as with the TREC L requests, may not be very useful in real terms, and that absolute performance even with a device generally found to be helpful can still be very low.

Table 3: Extract from Table 6

	Doc30			Rec30			AveP		
	L	M	V	L	M	V	L	M	V
UW	.04	.09	.15	.01	.05	.13	.01	.04	.09
CFW	.07	.15	.17	.04	.10	.17	.03	.07	.12

4.2 Relevance information

Information about term file incidence, though of some utility, is thus clearly only a very weak basis for estimating probability of relevance. The presumption is that as soon as more discriminating information about terms is available, and in particular any information about whether the documents in which a term is present are already actually known to be relevant or non-relevant, this will allow more accurate estimation. Thus for a more refined interpretation of the model we start (Robertson and Sparck Jones 1976)as in with the term incidence contingency table:

[4] alias inverse document frequency (IDF) weight

[5] This and subsequent tables in the text are extracts of selected results from the main Table 6, located in the Appendix to this part of the paper

	Relevant	Non-relevant	
Containing the term	r	$n-r$	n
Not containing the term	$R-r$	$N-n-R+r$	$N-n$
	R	$N-R$	N

where R is the number of relevant (liked) documents for this query
and r is the number of these containing the term.
(For simplicity, the suffix i has been ignored;
$r = r_i$ and $n = n_i$ are term-specific.)

Now, neglecting for the moment the question of whether our knowledge of the relevant documents for a query is complete, with the information just given we can estimate p_i and $\overline{p_i}$, namely (ignoring the suffix)

$$p = \frac{r}{R}$$

and

$$\overline{p} = \frac{n-r}{N-R}$$

We can then rewrite the term presence weighting function 5 as:

$$w = \log \frac{r(N-n-R+r)}{(R-r)(n-r)} \tag{7}$$

Different starting assumptions might lead to a slightly different formula (Robertson and Sparck Jones 1976)see e.g..

The relation between this weight and CFW is as follows. In the absence of relevance information, we may estimate $\overline{p}$ from the proportion of items in the collection that contain the term, that is n/N. The assumption here is that, in the context of the entire collection (N), the number of relevant documents (R) is likely to be small. For p, however, we have no evidence, and the simplest assumption would be that $p =$ constant. This pair of assumptions leads (Croft and Harper 1979) to a weight which is very similar to the collection frequency weight above, but not quite identical. While it is clear that the assumption that $p =$ constant is a very crude one, and almost certainly not generally true, it seems to be (both intuitively and experimentally) a reasonable starting point. A slight modification of the assumptions (Robertson and Walker 1997), in which p is not constant and which avoids some anomalies of the Croft/Harper model, leads to exactly the formula 6 above.

The problem of estimating p and $\overline{p}$ given some (small or large quantity of) relevance information is a general one which we need to develop. Any instantiation of the model for practical application requires further consideration of the estimation problem and of the information on which estimates may be based. Thus in practice we would normally be in the situation where, even if we know of some relevant documents, we wish to continue searching: i.e. we are assuming that we have not found all the relevant documents that would meet our need. The values in the central cells of the contingency table therefore cannot be taken as absolute and our estimates of document relevance when considering new items have to allow for uncertainty.

Estimation considerations give rise to a simple modification of formula 7 (Robertson and Sparck Jones 1976), namely to add 0.5 to all the central cells. We can then derive a specific term relevance weighting formula RW,

$$RW = \log \frac{(r+0.5)(N-n-R+r+0.5)}{(R-r+0.5)(n-r+0.5)} \tag{8}$$

with associated matching score

$$MS\text{-}RW = \sum_i \log \frac{(r_i+0.5)(N-n_i-R+r_i+0.5)}{(R-r_i+0.5)(n_i-r_i+0.5)}$$

These estimation considerations were not initially based on a Bayesian argument, as the form of the equation might suggest, but on minimising bias in the estimate of log-odds (Cox 1970).

This formula gives relatively higher weight to query terms that have a high relevant document incidence and low additional nonrelevant document incidence. It is well-behaved in extreme cases, unlike 7 which would be infinite under some conditions.

Table 4 illustrates the value of RW as a predictor of document relevance, compared with CFW, using the standard experimental method of computing weights from one half of a collection and applying them to the other, i.e. using all the relevant documents in one half to compute weights for searching the other. Comparing the runs labelled 'pred all' for these RWs with CFWs, and using Rec30 for all the collections, shows that the performance gain is typically very large, indeed often more than Dramatic, with the TREC V requests a somewhat surprising exception in showing no more than a Noticeable improvement. Overall, however, the performance difference is at least *RW pred all* $>$ *CFW*. The Doc30 results for the T741000X collection resemble the Rec30 ones, and all (including V) are statistically significant. But the amount of relevance information available in such tests may be quite large, and certainly larger than could normally be expected in the case where a user is online and is inspecting output from which information may be gathered to revise the query by modifying its term weights. So it is necessary to consider the effects of different amounts of relevance information, and useful to have reasonable grounds for believing that estimates based on rather little information, if this is of the right sort, may still be adequate, and hence that even where the incidence data is limited RWs can improve performance.

Table 4: Extract from Table 6

	Doc30			Rec30			AveP		
	L	M	V	L	M	V	L	M	V
CFW	.07	.15	.17	.04	.10	.17	.03	.07	.12
RW pred all	.29	.26	.19	.25	.24	.20	.17	.17	.13
RW pred top 3	.16	.21	.18	.13	.18	.18	.08	.12	.12
RW pred rel in 10	.21	.23	.18	.17	.20	.18	.12	.14	.12

In the earlier experiments reported in Sparck Jones (1979a), the amount of relevance information was systematically increased, reassuringly showing that performance correspondingly improved but also that relatively little information could still be of some value. Sparck Jones (1979b) also reported experiments comparing the use of only a few, randomly chosen relevant documents for prediction as against the full set. These tests also suggested that even a few relevant documents could be helpful, but all of these early tests used older and probably flattering methods of performance representation. The results shown under the label 'top 3' in Table 4, and for the older collections (just 'top 2' for Cranfield) in Table 7 also illustrate performance when only a few relevant documents are available but these are quality ones, namely the best matching ones.

Unfortunately, it is not sensible to define best matching in the same way for both older and new collections: for the older ones it was simply defined via the number of matching terms. For the T741000X collection, as the poor UW performance considered earlier implies, a more sophisticated as well as convenient means of identifying best matching documents is justified, and we therefore used the (up to) 3 relevant documents found in the top 100 ranks when searching with the combined weights defined in Section 4.7, Part 2. The figures for Rec30 for all the collections show that while predictive performance is poorer with only top 3 than with all relevant as a base for computing weights, it is also the case that top 3 performance is at least Noticeably better, and often more, than for CFW, i.e. $RW\ pred\ top\ 3 > CFW$ or more, except for the V form TREC requests. The same holds for Doc30 with the TREC collection, and the differences for TREC (even for the V requests) are statistically significant. It is worth also exploring the effect of alternative prediction bases for relevance weighting: thus as well as taking the top 3 relevant documents as a base, we have tested taking whatever relevant documents are found in the best matching 10 documents. The runs in Table 6 labelled 'rel in 10' can be taken as an informal simulation of the widespread page-based output display common in Web systems. The results show performance for $RW\ rel\ in\ 10$ similar to that for $RW\ top\ 3$ relative to CFW, though here the difference for V requests is not statistically significant.

Altogether, these runs show that even where only a few relevant documents are known the effects can be beneficial, while performance naturally improves as more information becomes available. This is also borne out by the more elaborate experiments we present in Section 4.9, Part 2, where the use of relevance data is combined with other information for weighting, and also by the TREC routing experiments examined in TR446 (1998).

Note that in all the experiments using small amounts of relevance information, e.g. those with top 3, we assume that all the documents not known to be relevant are non-relevant, and so contribute to the non-relevance probability. This is consistent with the argument in Croft and Harper (1979) but is not implied by it. One can also approach this point in a more sophisticated way – see Robertson and Walker (1997).

4.3 Retrospective relevance weights

As getting performance improvement in parallel with increasing relevance information suggests, we can relate this whole model interpretation very directly to the Probability Ranking Principle, and also very usefully for retrieval research purposes. Thus if we compute RW from the complete relevance information for a test collection and then apply the weighted queries retrospectively to the set of documents from which they were derived, the output ordering we get is in some sense the best that can be obtained for the given term sets of queries and documents. That is, we have optimised the descriptions. The utility of this retrospective application of RWs is thus in supplying a realistic upper performance bound, or *yardstick* (Sparck Jones 1975), against which actual performance based on prediction can be set.

The weight computation may be done either in absolute style directly exploiting the contingency table, as in Robertson and Sparck Jones (1976); or it may be done with the modified table using 0.5 as for predictive weighting. It can be argued that the former is more principled, and that the 0.5s introduced into the *RW* formula for estimation reasons are not appropriate for retrospective use; however they may still give the most appropriate upper-bound performance against which to compare other predictive methods. Table 5 shows retrieval performance when the *same* Half collection is used to compute and apply weights, using both absolute and qualified 'retro' formulae for older collections but only the qualified one for the T7410000X runs. The absolute formula is flagged by *. The data for the old collections for this version of the formula is however somewhat limited, so it can only be interpreted with caution as implying, not surprisingly, a higher level of attainable performance than the modified formula does. But more importantly, when performance for the modified formula is considered, on the values for Rec30 across all the collections, there is a great difference between the older collections and T741000X. For the older collections, yardstick performance is at least Strikingly better than predictive even when all the relevant documents are used, i.e. *RW retro* >>> *RW pred all*; the difference is therefore even larger when the comparison is made with prediction only from the best few. But with the TREC collection, regardless of request form, retrospective and predictive performance is the same, i.e. *RW retro* = *RW pred all* and for Doc30 as well as Rec30. This is not, however, at all surprising since with the older collections there are fewer relevant documents, while for TREC there are many in this specific comparison, so convergence between retrospective and predictive is rational. This is emphasised by comparing retrospective with predictive but from the small top 3 base: while for the V form requests we only have *RW retro* > *RW pred top* 3, for the others the difference is larger. The differences in Table 8 are again statistically significant. *RW retro* versus *RW pred rel in* 10 is similar, with statistically significant differences throughout, though informally there is no difference for the V form requests.

Table 5: Extract from Table 6

	Doc30			Rec30			AveP		
	L	M	V	L	M	V	L	M	V
RW retro	.30	.26	.19	.25	.24	.20	.18	.17	.13
RW pred all	.29	.26	.19	.25	.24	.20	.17	.17	.13
RW pred top 3	.16	.21	.18	.13	.18	.18	.08	.12	.12
RW pred rel in 10	.21	.23	.18	.17	.20	.18	.12	.14	.12

Of course where the test collection relevance judgements are not exhaustive, performance will not be strictly optimal. The technique is also limited to statements about queries with any given composition: it does not provide any guidance on how the term composition of a query might be modified to advantage. It can nevertheless provide a very useful picture of a collection's potential performance space, in the way that applying the Cluster Hypothesis to exhibit the separation of relevant and non-relevant documents (van Rijsbergen and Sparck Jones 1973) can also provide a background for assessing performance. We make further use of the yardstick to calibrate performance for the strategies described in the next sections.

4 Realism

The general question of what relevance information might be available in particular circumstances and how it might be used is only touched on here. For example, the effects studied in the older predictive tests described in Sparck Jones (1979a) were purely quantitative, i.e. they considered only the numbers of known relevant documents; and the experiments with the best matching 3 relevant documents described earlier are as much quantitative as qualitative. The older tests did not mimic the online searching case where the information available is biased (for good or ill) to documents that rank high in the proffered output. The tests with top 3 were somewhat nearer to real searching, in using best matching documents, but in taking a fixed number disregarded how many documents would have to be inspected to reach this. In all of these tests there was also only a single search iteration, where in reality there might be several. It is unfortunately extremely difficult to carry out proper tests to establish the value of iterative reweighting. With small test collections there are liable to be too few relevant documents left after the first cycle for performance effects from query reweighting to show. With larger collections and a good supply of relevance data this problem does not arise, but tests are unrealistic because they do not capture the effects of online interaction on user judgements. At the same time, iterative searching with real users does not deliver all the judgements needed for comparative purposes. We return to iterative searching later, along with other retrieval tasks where learning is involved.

Even with laboratory simulation, however, it is possible to be more realistic than the top 3 case allows: we examine some alternatives later in the context of additional indexing devices. Thus for the present we simply note that predictive relevance weighting with RW, i.e. in the basic form introduced in Robertson and Sparck Jones (1976), is of some value.

In Part 2 we continue with the treatment of basic retrieval data, and then proceed to elaborate the model. Appendix 1 to this part also includes results for the tests in Part 2.

References

Cooper, W. (1995) Some inconsistencies and misidentified modelling assumptions in probabilistic information retrieval. *ACM Transactions on Information Systems*, 13, 100–111.

Cox, D.R. (1970) *Analysis of binary data.* London: Methuen.

Croft, W.B. and Harper, D.J. (1979) Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35, 285–295.

Harman, D.K. (1997) Evaluation techniques and measures. The Fourth Text REtrieval Conference (TREC–4), (Ed. D.K. Harman), Special Publication 500-236, National Institute of Standards and Technology, Gaithersburg, MD, 1996, A-6–A-14.

Lancaster, W.F. (1969) MEDLARS: Report on the evaluation of its operating efficiency. *American Documentation*, 20, 119–142.

Maron, M.E. and Kuhns, J.L. (1960) On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, 7, 216–244.

Porter, M.F. (1980) An algorithm for suffix-stripping, *Program*, 14, 130–137.

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

van Rijsbergen, C.J. and Sparck Jones, K. (1973) A test for the separation of relevant and non-relevant documents in experimental retrieval collections. *Journal of Documentation*, 29, 251–257.

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

Robertson, S.E. and Belkin, N.J. (1978) Ranking in principle. *Journal of Documentation*, 34, 93–100.

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

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

Robertson, S.E. and Walker, S. (1997) On relevance weights with little relevance information. *Proceedings of the Twentieth Annual International ACM–SIGIR Conference on Research and Development in Information Retrieval*, New York: Association for Computing Machinery, 16–24.

Salton, G. (1975) *A theory of indexing.* Philadelphia, PA: Society for Industrial and Applied Mathematics.

Salton, G. and Buckley, C. (1988) Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24, 513–523.

Salton, G. and Buckley, C. (1990) Improving retrieval performance by relevance feedback, *Journal of the American Society for Information Science*, 41, 288–297, 1990.

Salton, G. and McGill, M.J. (1983) *Introduction to modern information retrieval.* Englewood Cliffs, NJ: Prentice Hall.

Sparck Jones, K. (1971) A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28, 11–21. (See also correspondence, 28(2), 164–165).

Sparck Jones, K. (1975) A performance yardstick for test collections. *Journal of Documentation*, 31, 266–272.

Sparck Jones, K. (1979) (1979a) Experiments in relevance weighting of search terms, *Information Processing and Management*, 15, 1979, 133–144.

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

Sparck Jones, K. (1999) (1999a) What is the role of NLP in text retrieval? In *Natural language information retrieval* (Ed. T. Strzalkowski), Dordrecht: Kluwer, 1–24.

Sparck Jones, K. (1999) (1999b) Summary performance comparisons: TREC–2 through TREC–7. In *The Seventh Text REtrieval Conference (TREC–7)*, Special Publication 500-242, National Institute of Standards and Technology, Gaithersburg, MD, B1–B6.

Sparck Jones, K. (1999) (1999c) Further reflections on TREC. *Information Processing and Management*, in press.

Sparck Jones, K., Walker, S. and Robertson, S.E. (1998) A probabilistic model of information retrieval: development and status. TR 446, Computer Laboratory, University of Cambridge (via http://www.cl.cam.ac.uk/).

Sparck Jones, K. and Webster, C.A. (1980) Research on relevance weighting 1976–1979, Computer Laboratory, University of Cambridge (also BL R&D Report 5553).

TREC (1992–1999): D.K. Harman (Ed.) *The First Text REtrieval Conference (TREC–1)*, Special Publication 500-207, National Institute of Standards and Technology, Gaithersburg, MD, 1993; *... Second ... (TREC–2)*. SP 500-215, 1994; *... Third ...(TREC–3)*, SP 500-225, 1995; *... Fourth ... (TREC–4)*, SP 500-236, 1996; Voorhees, E.M. and Harman, D.K. (Eds.) *... Fifth ... (TREC–5)*, SP 500-238, 1997; *... Sixth ... (TREC–6)*, SP 500-240, 1998; *... Seventh ... (TREC–7)*, SP 500-242, 1999.

Appendix: Detailed retrieval results and significance test data

Table 6 shows first, the run output for the TREC T741000X collection, i.e. the odd-numbered Half collection. Predictive relevance weights are computed from the even-numbered half, retrospective from the odd-numbered half. The TREC runs cover first the Long version requests, then Medium, then Very short.

Performance is given as Average Precision, labelled AveP; Precision at Document Cutoffs 5, 10, 15, 20, 30, 100, labelled P5, P10, etc; RPrec – i.e. Precision at rank corresponding to the number of relevant per query; Precision at Recall 30, labelled P30R; and Recall at rank 1000. Values for Document Cutoff at 30, labelled P30, are called Doc30 in the body of the paper; Precision at Recall 30, P30R, is called Rec30. The figures are simply truncated.

Table 7 shows corresponding figures, where available, for the old Cranfield C1400I, UKCIS U27000P and NPL NPL11500A collections; these are followed by alternative figures based on microaveraging.

Table 8 gives significance results for the TREC collection using the Wilcoxon signed ranks test, applied to Average Precision, Precision at Document Cutoff 30, and Precision at Recall 30, for all three versions of the requests. It also shows, for comparison, the value of the informal rating of performance differences using Precision at Document Cutoff 30.

Table 6: Retrieval run results

T741000X H collection, L, M and V requests

Long requests

	AveP	Doc5	Doc10	Doc15	Doc20	Doc30	Doc100	RPrec	Rec30	R1000
UW - term coord	.011	.059	.046	.041	.037	.035	.029	.026	.014	.164
CFW - coll freq wts	.031	.079	.076	.072	.068	.070	.059	.055	.044	.259
QACFW - query adj	.086	.120	.133	.141	.138	.145	.126	.124	.123	.442
RW - rel wts :										
retro	.181	.317	.326	.321	.313	.299	.237	.238	.253	.614
pred all	.174	.297	.313	.309	.302	.287	.230	.230	.245	.609
pred top 3	.084	.157	.165	.168	.164	.159	.133	.129	.126	.436
pred rel in 10	.116	.213	.221	.218	.216	.213	.170	.163	.168	.494
CW - comb wts	.225	.539	.505	.475	.449	.412	.282	.277	.316	.600
QACW - query adj	.320	.637	.585	.559	.534	.497	.356	.353	.440	.722
best pass	.329	.635	.583	.552	.528	.496	.357	.363	.443	.743
QACIW - comb iter wts, adj :										
retro *	.354	.676	.639	.612	.584	.543	.383	.387	.491	.750
retro	.354	.675	.637	.613	.583	.544	.383	.387	.490	.750
pred all	.346	.651	.626	.600	.576	.536	.379	.380	.481	.746
retro rel in 10	.338	.700	.633	.591	.558	.518	.364	.363	.460	.727
pred rel in 10	.334	.647	.614	.587	.558	.518	.365	.365	.462	.727
best pass	.350	.659	.603	.584	.565	.524	.372	.379	.472	.754
retro top 3	.337	.705	.636	.596	.563	.516	.364	.368	.458	.728
pred top 3	.332	.655	.615	.582	.557	.511	.364	.364	.457	.725
pred random 3	.330	.639	.607	.588	.552	.511	.365	.364	.458	.733
pred blind 10	.322	.628	.593	.564	.539	.496	.356	.353	.442	.717
QACIW + E - comb iter, adj, exp :										
retro rel in 10										
exp 32	.390	.800	.715	.668	.626	.576	.395	.400	.521	.753
pred rel in 10										
exp 72	.374	.733	.667	.640	.610	.569	.395	.394	.514	.754
exp 48	.374	.713	.663	.633	.612	.573	.395	.398	.512	.755
exp 40	.372	.723	.665	.636	.611	.571	.395	.395	.509	.756
exp 32	.370	.703	.654	.629	.608	.570	.394	.395	.506	.754
best pass	.389	.708	.675	.636	.609	.575	.400	.407	.515	.777
qterm emph 20/19	.365	.691	.646	.629	.603	.565	.393	.394	.503	.753
exp 24	.367	.696	.655	.630	.609	.564	.392	.393	.509	.747
exp 16	.355	.676	.644	.617	.594	.551	.383	.387	.496	.732
retro top 3										
exp 32	.388	.829	.725	.667	.626	.576	.393	.401	.521	.750
pred top 3										
exp 32	.351	.687	.638	.610	.579	.538	.377	.379	.478	.736
qterm emph 20/19	.349	.695	.650	.617	.584	.540	.378	.382	.482	.743
retro blind 10										
exp 32	.352	.648	.610	.587	.564	.523	.380	.377	.467	.753
pred blind 10										
exp 32	.345	.633	.602	.580	.556	.526	.378	.368	.474	.743

Table 6 (contd): Retrieval run results

Medium requests

	AveP	Doc5	Doc10	Doc15	Doc20	Doc30	Doc100	RPrec	Rec30	R1000
UW - term coord	.036	.116	.110	.101	.097	.093	.071	.065	.048	.284
CFW - coll freq wts	.074	.153	.157	.152	.149	.146	.113	.112	.104	.382
QACFW - query adj	.117	.180	.185	.191	.190	.187	.159	.163	.169	.485
RW - rel wts :										
retro	.174	.293	.289	.272	.266	.264	.211	.220	.242	.600
pred all	.168	.275	.271	.263	.258	.255	.205	.211	.236	.595
pred top 3	.124	.209	.225	.226	.223	.213	.171	.177	.184	.501
pred rel in 10	.139	.208	.223	.233	.233	.227	.182	.187	.201	.522
CW - comb wts	.226	.513	.481	.459	.437	.397	.273	.281	.320	.598
QACW - query adj	.269	.536	.524	.509	.486	.442	.310	.320	.374	.651
best pass	.282	.537	.513	.493	.473	.445	.314	.326	.385	.671
QACIW - comb iter wts, adj :										
retro *	.300	.603	.567	.547	.525	.484	.338	.348	.427	.684
retro	.300	.601	.564	.547	.525	.484	.339	.348	.427	.685
pred all	.296	.592	.559	.539	.522	.483	.335	.345	.421	.681
retro rel in 10	.287	.615	.563	.535	.505	.466	.325	.333	.402	.659
pred rel in 10	.282	.580	.541	.523	.497	.456	.324	.332	.399	.659
best pass	.300	.565	.538	.521	.501	.469	.329	.341	.412	.684
retro top 3	.283	.615	.561	.532	.505	.464	.320	.331	.396	.658
pred top 3	.276	.556	.529	.514	.494	.455	.319	.327	.389	.655
pred random 3	.283	.576	.538	.516	.496	.457	.323	.335	.402	.666
pred blind 10	.277	.549	.523	.510	.491	.450	.318	.327	.393	.654
QACIW + E - comb iter, adj, exp :										
retro rel in 10										
exp 24	.364	.787	.683	.639	.595	.540	.365	.380	.492	.708
pred rel in 10										
exp 32	.335	.656	.617	.587	.562	.518	.358	.364	.469	.714
exp 24	.336	.648	.616	.586	.560	.517	.362	.368	.471	.713
best pass	.353	.645	.617	.594	.570	.522	.370	.378	.476	.737
qterm emph 20/19	.337	.651	.613	.587	.566	.526	.365	.367	.467	.720
exp 16	.335	.640	.607	.587	.563	.522	.364	.369	.469	.715
retro top 3										
exp 24	.360	.819	.703	.646	.608	.550	.366	.382	.483	.709
pred top 3										
exp 24	.324	.667	.623	.585	.555	.514	.350	.359	.447	.704
qterm emph 20/19	.328	.659	.629	.589	.566	.519	.354	.361	.458	.711
retro blind 10										
exp 24	.317	.575	.549	.521	.502	.473	.348	.353	.434	.710
pred blind 10										
exp 24	.318	.589	.571	.551	.527	.494	.352	.353	.452	.714

Table 6 (contd): Retrieval run results

Very short requests

	AveP	Doc5	Doc10	Doc15	Doc20	Doc30	Doc100	RPrec	Rec30	R1000
UW - term coord	.087	.165	.159	.160	.158	.154	.125	.126	.131	.404
CFW - coll freq wts	.116	.177	.167	.170	.173	.174	.154	.157	.167	.475
RW - rel wts :										
retro	.134	.189	.183	.185	.186	.193	.171	.173	.196	.532
pred all	.132	.183	.180	.180	.182	.189	.169	.173	.196	.528
pred top 3	.121	.179	.169	.176	.179	.179	.162	.163	.178	.491
pred rel in 10	.124	.180	.169	.176	.178	.180	.164	.164	.183	.500
CW - comb wts	.244	.501	.473	.451	.431	.403	.283	.291	.343	.602
QACW - query adj	.244	.501	.473	.451	.431	.403	.283	.291	.343	.602
best pass	.248	.485	.466	.452	.430	.399	.283	.288	.335	.609
QACIW - comb iter wts, adj :										
retro *	.269	.559	.528	.500	.474	.436	.309	.310	.376	.642
retro	.269	.555	.528	.501	.475	.436	.309	.310	.376	.642
pred all	.265	.545	.529	.494	.470	.433	.306	.305	.372	.641
retro rel in 10	.253	.535	.505	.472	.452	.418	.293	.296	.357	.609
pred rel in 10	.252	.527	.499	.470	.449	.418	.291	.297	.356	.608
best pass	.259	.509	.498	.479	.453	.418	.292	.300	.353	.618
retro top 3	.253	.539	.513	.478	.454	.420	.292	.297	.355	.611
pred top 3	.248	.515	.489	.468	.448	.416	.289	.294	.352	.607
pred random 3	.257	.529	.507	.474	.456	.426	.296	.298	.363	.623
pred blind 7	.243	.492	.475	.452	.430	.403	.284	.289	.343	.591
QACIW + E - comb iter, adj, exp :										
retro rel in 10										
exp 16	.307	.715	.622	.577	.541	.488	.321	.331	.434	.611
pred rel in 10										
exp 24	.265	.575	.532	.509	.484	.436	.296	.300	.368	.608
exp 16	.260	.547	.516	.495	.471	.433	.295	.294	.360	.602
best pass	.273	.553	.519	.496	.479	.435	.302	.303	.368	.624
qterm emph 20/19	.266	.567	.521	.499	.476	.441	.298	.299	.366	.609
retro top 3										
exp 16	.294	.769	.653	.587	.539	.482	.314	.329	.397	.620
pred top 3										
exp 16	.252	.587	.535	.498	.476	.431	.291	.298	.346	.604
qterm emph 20/10	.264	.591	.546	.502	.481	.442	.301	.310	.361	.621
retro blind 7										
exp 16	.241	.487	.461	.443	.421	.394	.282	.279	.338	.588
pred blind 7										
exp 16	.241	.495	.469	.449	.426	.395	.279	.278	.336	.579

Table 7: Retrieval run results, old collections
C1400I H, U27000P H, N11500A H collections (e = guestimated).

	AveP	Doc5	Doc10	Doc20	Doc30	Doc100	Rec30	R1000
C1400								

UW	.29						.39	
CFW	.30						.40	
RW retro *	.53						.68	
retro	.43						.56	
pred all	.35						.47	
pred top 2	.34						.43	
U27000Pb								

UW	.30						.42	
CFW	.32e						.45e	
RW retro *	.55e						.75e	
retro	.42						.60	
pred all	.37						.54	
pred top 3	.36						.51	
N11500A								

UW	.20	.27	.24	.18		.07	.29	
CFW	.22e						.33e	
RW retro *	.44	.46	.37	.27		.09	.59	
retro	.37	.44	.36	.27		.09	.54	
pred all	.31	.39	.32	.23		.09	.45	
pred top 3	.27	.36	.29	.21		.08	.40	

Table 8: Significance test results, selected runs, T741000X H collection

```
Wilcoxon signed ranks test: - means < 1.96, . means 1.96 - 2.33, + means > 2.33
 1.96 and 2.33 correspond to 2.5% and 1% significance levels on one-tail test
All request versions:  group of three tests is for AveP, Doc30, Rec30
Run labels: r/10 = rel in 10; e16, 32 etc = exp by 16, 32 etc
            q em = qterm emph; b pa = best passage
nV, L etc = except V, L request form in informal comparison
```

			Infml	L	M	V
CFW	vs	UW	>	+ + +	+ + +	+ + +
RW pred all		CFW	>	+ + +	+ + +	+ + +
RW pred top 3		CFW	> nV	+ + +	+ + +	+ . +
RW pred r/10		CFW	>>>> nV	+ + +	+ + +	+ - +
RW retro		RW pred all	>	+ + +	+ + -	+ . -
RW retro		RW pred top 3	>	+ + +	+ + +	+ + +
RW retro		RW pred r/10	>> nV	+ + +	+ + +	+ + +
CW		CFW	+>>>>	+ + +	+ + +	+ + +
CW		RW retro	>>>	+ + +	+ + +	+ + +
QACW		CW	>> nV	+ + +	+ + +	- - -
QACIW retro		RW retro	+>>>>	+ + +	+ + +	+ + +
QACIW pred all		RW pred all	+>>>>	+ + +	+ + +	+ + +
QACIW pred top 3		RW pred top 3	+>>>>	+ + +	+ + +	+ + +
QACIW pred r/10		RW pred r/10	+>>>>	+ + +	+ + +	+ + +
QACIW retro *		QACIW retro	=	- - -	- - -	- - -
QACIW pred all		QACW	>	+ + +	+ + +	+ + +
QACIW pred top 3		QACW	=	+ . +	+ . -	- + -
QACIW pred r/10		QACW	=	+ + +	+ . +	+ + +
QACIW pred blind		QACW	=	- - -	. - -	- - -
QACIW retro		QACW	>	+ + +	+ + +	+ + +
QACIW retro		QACIW pred all	=	+ + +	+ - +	+ - -
QACIW retro		QACIW pred top 3	>	+ + +	+ + +	+ + +
QACIW pred all		QACIW pred top 3	>	+ + +	+ + +	+ + +
QACIW pred all		QACIW pred r/10	>	+ + +	+ + +	+ + +
QACIW pred all		QACIW pred rand 3	=	+ + +	+ + +	+ - +
QACIW retro top 3		QACIW pred top 3	=	+ - -	+ . .	+ - -
QACIW retro r/10		QACIW pred r/10	=	+ - -	+ + -	- - -
QACIW+E pred top 3		QACIW pred top 3	> nV	+ + -	+ + +	- - -
QACIW+E pred r/10		QACIW pred r/10	> nV	+ + +	+ + +	- - -
QACIW+E pred blind		QACW	> nV	+ + +	+ + +	- - -
QACIW+E pred blind		QACIW pred blind	> nV	+ + +	+ + +	- - -
QACIW+E retro top 3		QACIW retro top 3	>	+ + +	+ + +	+ + -
QACIW+E retro r/10		QACIW retro r/10	>	+ + +	+ + +	+ + +
QACIW+E retro top 3		QACIW+E pred top 3	>>	+ + +	+ + +	+ + +
QACIW+E retro r/10		QACIW+E pred r/10	> nL	+ - -	+ - -	+ + +
V QACIW+E pred r/10 e24		QACIW+E pred r/10 e16	=			- - .
M QACIW+E pred r/10 e32		QACIW+E pred r/10 e16	=		- - -	
L QACIW+E pred r/10 e48		QACIW+E pred r/10 e16	>	+ + +		
L QACIW+E pred r/10 e72		QACIW+E pred r/10 e16	=	+ + +		
QACIW+E pred top 3 q em		QACIW+E pred top 3	=	- - -	+ - +	+ + +
QACIW+E pred r/10 q em		QACIW+E pred r/10	=	- - -	- - -	+ . .
QACIW+E pred r/10		QACIW+E pred blind	>	+ + +	+ . -	. + -
QACIW pred r/10 b pa		QACIW pred r/10	=	+ - -	+ - -	. - .
QACIW+E pred r/10 b pa		QACIW+E pred r/10	=	+ - -	+ - -	+ - -