A probabilistic model of information retrieval : development and status

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[1]

August 1998

[1]This work was done while Robertson and Walker were at the Department of Information Science, City University, London.

Abstract

The paper combines a comprehensive account of the 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.

Contents

1 Introduction **3**

2 Foundations **5**
2.1 Introduction 5
2.2 Probability of relevance 5
2.3 Information about documents 8
2.4 Formal model 8
2.5 Independent attributes 9
2.6 Term presence and absence 10

3 Test collections and measures **11**
3.1 Data sets 11
3.2 Performance measures 13

4 Data **18**
4.1 Term incidence 18
4.2 Relevance information 19
4.3 Retrospective relevance weights 22
4.4 Realism 23
4.5 Term frequencies and weighting 23
4.6 Document lengths and weighting 25
4.7 Instantiations 26
4.8 Long queries 27
4.9 Frequency experiments 27

5 Elaboration **30**
5.1 Query expansion from relevance information 30
5.2 Selective or massive expansion? 32
5.3 Expansion without relevance information 33
5.4 Term cooccurrences 34
5.5 Term phrases 35
5.6 Document levels: passages 38
5.7 Final note 39

6 Environment conditions **41**
6.1 Document properties 41
6.2 Request properties 41
6.3 Languages 42

7 Tasks **43**
7.1 Interactive searching 43
7.2 Text extraction 44
7.3 Document summarisation 44
7.4 Routing and filtering 44
7.5 Absolute scores 45
7.6 Indexing and category assignment 46

8 Training **48**
8.1 Learning about the information need 48
8.2 Learning about documents 49
8.3 Learning about terms 49
8.4 Regression 50
8.5 Data exploration 50
8.6 Some pitfalls of training 50

9 Comparisons **52**
9.1 The vector space model 52
9.2 Probabilistic indexing and a unified model 52
9.3 Dependency 53
9.4 Logical information retrieval 53
9.5 Networks 53
9.6 Regression 54

10 Assessment **56**
10.1 Test summary and review 56
10.2 The model's status in IR theory 57
10.3 Open issues 58

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.

As what we will for convenience label the City model has evolved to accommodate test findings and to meet an increasing range of retrieval circumstances since its initial publication in Robertson and Sparck Jones (1976), the way we present the model in this paper has some historical reference. However 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 following Section 5, Elaborations, then extends the model to more complex cases as illustrated by, for instance, query expansion. In Section 6, Environment conditions, we examine the consequences of different variable values, for example long documents as opposed to short ones, for the form and application of the model. In the Foundations section simplifying assumptions were made about the shape of retrieval data, and hence about what constraints (or rather lack of constraints) these impose on model interpretation and ultimately on system behaviour. The Elaborations and Conditions sections considers the richer model interpretations that are needed to exploit more productive possibilities or respond to more challenging situations. Then, since in the sections so far we have considered the model only in the context of the default, normal retrieval task, namely the single-occasion or so-called adhoc search, in Section 7, Tasks, we examine the form and application of the general model for other tasks or subtasks within the wider retrieval area, for instance in user support in inter-

active searching or in routing. Section 8, Training, discusses contexts for system adaptation and the forms this takes in the model.

The general pattern we will follow in these sections is to present and motivate the essential relevant 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 tets, and refer to other experiments done by City under the TREC Programme, since these extend the range of comparisons over time and test conditions. As mentioned earlier, our aim is to provide a coherent and integrated account of work done over a long period,. We will therefore focus on its major elements and refer to other publications for amplifying details. Further, up to the end of Tasks we will concentrate on City work, referring to others only where necessary for local purposes. 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, and on present status of the City model, ending with a final Open Issues discussion of unresolved problems and future challenges.

The set of experiments covered in the paper is large, and they are referred to at many points in 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.

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 item 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.

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

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)
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
CIW	Combined iterative weight	13
$QACW$	Query-adjusted combined weight	14
$QACIW$	Query-ajusted combined iterative weight	15
OW	Offer weight	16
Tuning parameters		
k_1	Effect of term frequency	9
b	Effect of document length	10
K	Combination of k_1, b and document length	11

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 to 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 – see Section 7 – but such cases should be seen as special cases or simple extensions of the general one.

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

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 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 log-odds rather than the probability:[2]

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

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

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

[2] Log-odds can be derived from probability by an order-preserving transformation, and thus satisfy the Probability Ranking Principle given above. It is also useful to make this transformation for other reasons which will become apparent below.

role analogous to van Rijsbergen (1979)'s 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

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

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,

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

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, 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

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

Now if we define:

$$W(A_i = a_i) = \log \frac{P(A_i = a_i|L)P(A_i = 0|\overline{L})}{P(A_i = a_i|\overline{L})P(A_i = 0|L)} \tag{3}$$

it follows from equation 2 that

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

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 1992-1997) 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) and (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 historic 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 (following Porter (1980)), 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. 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. [3] 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 Jones (1998b) and (1998c) 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 contolled 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, and for this we have applied an experimental paradigm in which document sets are divided into similar halves, e.g. even- and odd-numbered respectively, with one used for weight computation and the other for weight application. We refer to the latter test set as Half, H. This naturally implies that all runs are actually on the smaller 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,

[3] 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.

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 (1992-1997), 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, 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 constants 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). [4]

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 (1998b) and (1998c). 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. We therefore also include a subsidiary table for these, showing Precision at standard Recall based on microaveraging across matching scores, to amplify somewhat the single number data drawn from ranked output (the figures are, however, liable to exaggerate performance differences, especially at low recall).

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 (1992-1997), and through these to work done elsewhere, as discussed in Section 9.

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

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

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 6 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. 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 Table 5. 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 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).

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 (1993-7)

   TREC 'T741000X'
     741856 documents in news, computing, official publications and energy
       represnted 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
C1400I	1400	2683	29.9	225	7.9	1614	7.2
H	700					780	3.5
U27000P	27361	n/a	n/a	75	18.3	3739	49.9
H	13681					1902	25.4
N11500A	11429	7491	20.0	93	7.2	2083	22.4
H	5715					1061	11.4
T741000X	741856	1290709*	129.9	150	L 32.6	37819	252.1
					M 10.3		
					V 4.0		
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.

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 focussing on the query also has a wider justification which reinforces the use of a probabilistic model. First, it fits with the postcoordinate philosophy that underpins modern retrieval systems, where initial document descriptions are open and are then closed into final ones by query matching. Second, it fits with the shift from file time to search time indexing as a logical as much as 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 more importantly, 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, because the number of relevant documents for a query is normally low by comparison with file size, the presence of a rare query term in a document is usually a better predictor of relevance than that of a 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)[5] 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. 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 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 6 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.

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 (as in Robertson and Sparck Jones (1976)) with the term incidence contingency table:

	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.)

[5] alias inverse document frequency (IDF) weight

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 (see e.g. Robertson and Sparck Jones (1976)).

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. A slight modification of the assumptions (Robertson and Walker 1997), to avoid 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

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

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 5 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. SSS 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.

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' (or just 'top 2' for Cranfield, as a smaller collection) in Table 5 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 below. 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. $RW\ top\ 3\ >\ CFW$ 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. 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 later in Section 4.9, where the use of relevance data is combined with other information for weighting, and by the TREC routing experiments we examine more fully in the Tasks section, 7.

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 *RW*s 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 top* 3 $>$ *RW pred top* 3, for the others *RW retro top* 3 $>>>$ *RW pred top* 3. The differences in Table 6 are again statistically significant.

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.

4.4 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, when we consider 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.

4.5 Term frequencies and weighting

Term incidence data is the most salient for exploitation; and it is important because at least in its most basic form, without relevance annotation, it comes with any file. We can moreover hope, if not expect, to be able to have some relevance incidence data to use as well. But there is other information about term behaviour which may also be available and useful for interpreting the model. In particular, the information used so far assigns different values to terms only according to their distribution across document files, and does not, for a particular term, distinguish one file document containing it from another.

The natural further interpretation of the model is therefore to exploit information about term frequencies within documents, if this can be supplied in initial descriptions. We require some way of modelling term frequency, and (as with the previous data) of relating this variable to relevance. Term frequencies within documents have in the past been modelled using Poisson distributions (Harter 1975); the particular model proposed here is a development of Harter's model (Robertson and Walker 1994). As before, the assumptions from which the model starts are clearly over-simplifications, but may help us develop a useful approach to retrieval.

We assume first that each term is associated with a *topic* (the idea or concept conveyed by that term), and that a document may be *about* the topic or not. That is, for each such topic, there is one set of documents about it and another (its complement in the file) that is not about it. We also allow, however, that the use of the term in text has some unpredictability about it; an author writing about the topic in question may use the term to a greater or lesser extent. Furthermore, an author not writing about this particular topic may refer to it in passing.

Since we do not know which documents are about the topic and which not, the distribution of within-document term frequencies that we observe is a mixture of two distributions, one in each of the two sets. The basic assumption is that both these distributions are Poisson. A Poisson distribution is defined by a single parameter, the mean: but the two distributions are clearly likely to have different means. The Poisson assumptions also really only make sense if all documents are of equal length. We assume this for now, and return to the document length question later.

The property of being about the topic or concept referred to by a term is called *eliteness* for the term in (Harter 1975). We denote this E, so E_i means "elite for term t_i", and $\overline{E_i}$ means "not elite for term t_i". TF_i is the frequency of term t_i in the document under consideration. As before, we may drop the suffix. The Poisson distribution assumptions will give us formulae for $P(TF|E)$ and $P(TF|\overline{E})$, in terms of each of the two Poisson means. The same formulae will cover the case $TF = 0$, i.e. the term is absent.

We can also define the probabilities for eliteness given likedness, namely $P(E|L)$ and $P(E|\overline{L})$. The basic assumption (Robertson, van Rijsbergen and Porter 1981) is now that TF depends directly on eliteness only, so that the relationship between TF and likedness is through eliteness. This relationship is expressed by means of two equations, one involving L:

$$P(TF|L) = P(TF|E)P(E|L) + P(TF|\overline{E})P(\overline{E}|L)$$

and a second, similar one involving $P(TF|\overline{L})$.

Now referring back to formulae 3 and 4 in Section 2, we see that the event $A_i = a_i$ may be interpreted as $A_i = TF_i$ or just as TF_i, and $A_i = 0$ as "term t_i absent". We may therefore express $W(TF_i)$ in formula 3 as a function of the Poisson distribution parameters and such quantities as $P(E_i|L)$.

The resulting formula is complex. This is not so much a question of algebraic complexity (although that is the case), as complexity of interpretation and estimation. Since eliteness is as invisible as relevance, neither the Poisson parameters nor the probabilities such as $P(E|L)$ can in general be directly estimated. However, in Robertson and Walker (1994), the behaviour of this formula is examined, and a much simpler formula which has similar behaviour is proposed and tested. The simpler formula is as follows

$$W(TF_i) = \frac{TF_i(k_1 + 1)}{k_1 + TF_i}\ w_i \tag{9}$$

Here w_i is the usual presence weight of term t_i (formula 5); k_1 is a constant, discussed below. The behaviour of this simple formula, mimicking the complex one, is that (a) it is zero for $TF_i = 0$, (b) it increases monotonically with TF_i, and (c) it has an asymptotic limit. When $TF_i = 1$, the weight is just the usual presence weight w_i; additional occurrences of t_i increase its contribution to the score, but there is an absolute limit on how much they can add. This asymptotic limit should in fact be the weight that would be associated with eliteness (if we could *know* whether a document was elite for the term or not). However, using the usual presence weight w_i in the formula gives us a plausible approximation.

The constant k_1 determines how much the weight reacts to increasing TF. If $k_1 = 0$, the weight reduces to the term-presence weight only; if k_1 is large, the weight is nearly linear in TF. It may be regarded as a tuning constant, to be adjusted after experimentation with the particular database. In TREC, we have found values in the range 1.2–2 to be effective. This small range implies that the effect of TF is highly non-linear, i.e. after say 3 or 4 occurrences

of a term the impact of additional occurrences is minimal. (See Section 8 for a discussion of the issues around discovering such values.)

Exploiting Poisson ideas thus means we have a way not only of bringing two separate types of information about terms and documents together, but of capturing the significance of different frequencies for terms in a single document in relation to term behaviour across the file: a document has a higher probability of relevance not simply if a term is frequent in it, but is unusually frequent given the number of documents in which it appears. Further, all of the argument works with any of the earlier instantiations of w_i, for example the collection frequency weight CFW or the relevance weight RW (formulae 6 and 8 respectively). These instantiations are discussed in the next section, after considering document length.

4.6 Document lengths and weighting

While it is wholly plausible to take term frequencies into account, the development of the formulae so far has tacitly assumed that all documents are the same length; and indeed the Poisson approach assumes constant length. In practice documents are not merely not all the same length, they may vary widely in length; and it is clear that one document containing a term t should not be preferred to another because t is more frequent in the former than than the latter if this is simply because the first document is twice as long as the second. Of course documents may vary in length for different reasons. But if we make one assumption, again of a rather simple but not unreasonable kind given the nature of the retrieval task, we can extend our model interpretation to deal with varying document length.

The simplest assumption is that where there are two documents about the same topic but of different lengths, this is just because the longer is more wordy. When closely examined from a linguistic point of view, as embodying a model of discourse, this is a very crude assumption: it implies wordiness is attributable merely to repetition rather than greater elaboration etc. But as retrieval normally deals with topic description at a a fairly general level, it may be sufficient to equate refinement with prolixity. On this assumption it is appropriate to extend the model interpretation to normalise term frequency by document length.

A simple normalisation (dividing TF by DL) would have the effect of giving the same score to a document of length DL in which a term t occurs TF times, as to a document of length $2DL$ in which the same term occurs $2TF$ times. But the crudity of the assumption is likely to lead to a bias in the above normalisation. That is, the $2DL$ document is unlikely to require a *smaller* score than the DL document, and it may be justifiable to give it a larger one (e.g. if wordiness suggest greater elaboration rather than just repetition). The slightly more complex normalisation suggested below, a mixture of no normalisation at all and the above simple normalisation, allows for this.

Another consideration is how document length may be measured. One could make many suggestions (e.g. word types or tokens, with or without stopwords, or simply characters). It probably does not matter much which is used, but it is appropriate to introduce some uniformity of scaling by relating document length to the length of an average document (in the same units). This will ensure that a document of average length will get the same score after document length normalisation as it had before.

The simple normalisation factor would therefore be $NF = \frac{DL}{AVDL}$. The mixed normalisation factor would be $NF = ((1-b) + b\frac{DL}{AVDL})$, with another tuning constant b, discussed further below. Considering the TF component of the TF formula 9 above, after normalisation

we have

$$\frac{\frac{TF_i}{NF}(k_1+1)}{k_1+\frac{TF_i}{NF}} = \frac{TF_i(k_1+1)}{k_1 * NF + TF_i}$$

Hence the weight becomes

$$W(TF_i) = \frac{TF_i(k_1+1)}{k_1 * ((1-b) + b\frac{DL}{AVDL}) + TF_i}\ w_i \tag{10}$$

If the new tuning constant b is set to 1, the simple normalisation factor is used (corresponding to an assumption of pure verbosity). Smaller values reduce the normalisation effect. Experiments with the TREC collection suggest a value of around $b = 0.75$ is good [6].

In order to simplify the presentation of equation 10, we replace $k_1 * NF$ with K, as follows:

$$W(TF_i) = \frac{TF_i(k_1+1)}{K + TF_i}\ w_i \tag{11}$$

where $K = k_1 * ((1-b) + b\frac{DL}{AVDL})$

The assumptions made here about the nature of document length differences are of course not the only possible ones: the most obvious other situation is where length relates to multi-topicality. We return to this in Section 5.

4.7 Instantiations

The last weighting function, 11, encapsulates the way terms gain value within the probabilistic framework from different types of information. It may be instantiated with or without relevance information, as suggested in the previous section. We will give these two instantiations different names. First, when using just the collection frequency weight CFW (equation 6), without relevance information, the combined weight is

$$CW = \frac{TF(k_1+1)}{K + TF}\ \log\frac{N}{n} \tag{12}$$

(with a corresponding matching score *MS-CW*, the sum of the combined weights of the matching terms).

Second, with relevance information, using the relevance weight RW (equation 8), the combined iterative weight (named to mark the fact that getting and using relevance information is an essentially iterative process) is

$$CIW = \frac{TF(k_1+1)}{K + TF}\ \log\frac{(r+0.5)(N-n-R+r+0.5)}{(R-r+0.5)(n-r+0.5)} \tag{13}$$

(again with a corresponding matching score *MS-CIW*).

In both of these weighting formulae, we have the two tuning constants k_1 and b, as already explained.

[6] We use the name b rather than the more obvious k_2 for compatibility with other papers, where b is used for this purpose while k_2 is used for something else.

4.8 Long queries

The weighting formulae just given would be directly suited to many retrieval applications, namely those where initial queries are quite simple, consisting just of a few words. However, a further refinement of the formulae may be devised to cover the possibility that terms are repeated in the query itself, as would naturally occur if a previously-known document was used as a starting query The theoretical basis for the refinement is very similar to that for the inclusion of within-document term frequency: a model based on a mixture of Poisson distributions, but applied to the set of queries rather than to the set of documents.

In principle, this leads to a QTF component similar to the TF component above, but with its own k tuning constant analogous to k_1. Experiments with TREC suggest that the higher the value of such a k, the better (in contrast with the TF component). But as indicated in Section 4.5, large k corresponds to an almost-linear function of term frequency, in this case of QTF. The formulae presented below are therefore based on a simple linear relationship, and no additional tuning constant is needed.

As before, there are two formulae, representing the situation without and with relevance information. The two weighting functions are called $QACW$ (query adjusted combined weight) and $QACIW$ (query adjusted combined iterative weight):

$$\begin{aligned} QACW &= CW * QTF \\ &= \frac{TF(k_1+1)}{K+TF}\ QTF\ \log\frac{N}{n} \end{aligned} \tag{14}$$

and

$$\begin{aligned} QACIW &= CIW * QTF \\ &= \frac{TF(k_1+1)}{K+TF}\ QTF\ \log\frac{(r+0.5)(N-n-R+r+0.5)}{(R-r+0.5)(n-r+0.5)} \end{aligned} \tag{15}$$

Again the weights should be summed over matching terms for the matching scores *MS-QACW* and *MS-QACIW* respectively.

4.9 Frequency experiments

The introduction of term frequencies leads to a large number of new performance comparisons, when both one-off and iterative searching are taken into account and term frequencies in queries as well as documents are covered. These comparisons only apply directly to the TREC T741000X collection results shown in Table 5, but have indirect connections with the older collections in showing the need to respond to the consequences of using full text as opposed to short document surrogates.

The first comparison is simply one without the use of relevance information, i.e. between plain CFW and full CW weights, and without query adjustment. Table 5 shows very large performance improvements regardless of whether Long, Medium or Very short queries are involved, and for both Rec30 and Doc30. The gain is more than Dramatic, i.e. $CW + >>>> CFW$, and the significance tests are correspondingly good. Thus these TREC data experiments both support the early theoretical arguments for using term frequency information, which has also been a consistent feature of the SMART system and a general strategy within the TREC Programme, and demonstrate its real value with full text, which earlier experiments only with abstracts or small sets of longer texts could not convincingly

do. It is also noteworthy that simply incorporating term frequencies (suitably normalised), for these full text cases, is not merely better than the most favourable predictive *RW* relevance weighting, using all documents, but than retrospective relevance weighting. Thus the difference, for either performance measure, is at least Striking, i.e. *CW* >>> *RW retro*, and the difference is significant.

When query adjustment, *QA*, is applied in *QACW* versus *CW* there is, not surprisingly, further gain for the Long requests, with *QACW* >>>> *CW* on either measure, and also for Medium where *QACW* >> *CW*, but nothing for Very short requests. The informal comparison is borne out by the significance test results. But the lack of improvement for the V requests is hardly surprising with only 4 terms per request; and since the general evidence is in favour of rather than against using frequency information for queries when it is available we have adopted query adjustment as the default in tests with other strategies.

Thus if we now consider the combination of term frequency and relevance information, under the label *QACIW* in Table 5, and compare the corresponding runs for *QACIW* and for *RW*, where relevance information alone is used, we find very marked gains from using frequencies. Thus for the retrospective case the performance improvement on either measure is more than Dramatic, i.e. *QACIW retro* + >>>> *RW retro*, and the same holds for the predictive cases using all or top 3 relevant documents: i.e. *QACIW pred all* + >>>> *RW pred all*, and *QACIW pred top* 3 + >>>> *RW pred top* 3. These large differences are also independent of request form, and are highly significant. Thus these runs very clearly show the advantages of a more refined treatment of the primary data about term occurrences within documents, even when relevance information is also available.

At the same time, checking runs for the TREC collection in order to compare the two treatments of the yardstick formula, without and with 0.5, for *QACIW* give rather different results from the analogous ones for the older collections using *RW* (there is little point in trying both *RW* yardsticks for TREC when performance for *RW* is so low compared with that for *CIW*). Thus for T741000X, *QACIW retro* ∗ = *QACIW retro*. But this is not surprising, given the much larger numbers of relevant documents available for prediction.

It is more important to compare *QACIW* with *QACW* for the effect of relevance information within the combined weight context and also, given the potential advantages of iterative weights, to consider more carefully the effects of variations in the amount and/or quality of relevance information. The runs shown in Table 5 confirm the value of extensive relevance information, with both Rec30 and Doc30 measures, though the gains are perhaps not as large as might be expected: thus in the weakest case, the improvement is no more than Noticeable, i.e. *QACIW pred all* > *QACW*. Further, when only the top 3 relevant documents are used for prediction, there is no gain at all in most cases: thus the *general* conclusion for the top 3 runs cannot be other than *QACIW pred top* 3 = *QACW*. The difference in comparative behaviour for *QACIW pred all* and *pred top* 3 is supported by the significance test results.

Indeed even with retrospective relevance weighting, the gains from relevance information are not enormous. Thus while the difference between having the best relevance information and not having any at all is material for the Long and Medium request forms, this is not an impressive difference, and the difference is smaller for the Very short requests, so the overall picture is only *QACIW retro* > *QACW*. Further, as all the foregoing implies, there is also little difference between *QACIW* retrospective and *QACIW* predictive, whether using all relevant or only top 3 for the latter. Thus *QACIW retro* = *QACIW pred all*, though *QACIW retro* > *QACIW pred top* 3. Again, this informal view is endorsed by the significance tests.

Altogether these results suggest that while there are large performance gains to be made from using term frequencies, those to be further made from adding in relevance information may not be so large, certainly on the rather strong view we are taking of worthwhile performance differences. But it is still useful to explore the effect of alternative prediction bases for relevance weighting when this is applied in the context of term frequency-based indexing, and indeed to do this more fully than before. 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 5 labelled 'rel in 10' can be taken as an informal simulation of the widespread page-based output display common in Web systems. We have also compared these relatively favourable cases with using 3 relevant documents drawn at random from the full set ('rand 3'), replicating the purely quantitative base studied in earlier experiments, but in a more stringent way through drawing a smaller sample from the larger TREC relevance set. We consider later, in Section 5.3, the case where best matching documents on a first pass are deemed relevant, whether they actually are or are not so (the runs labelled 'blind').

The results given in Table 5 show that, regardless of request form and on both performance measures, while the predictive all case is Noticeably better than the other prediction bases except for the Very short requests, there is no difference between the others. More specifically, if we exclude the slightly unexpected results for 'rand 3', which performs better than might be expected for the Very short requests, we find that $QACIW\ pred\ all\ >\ QACIW\ pred\ rel\ in\ 10$, as we earlier found $QACIW\ pred\ all\ >\ QACIW\ pred\ top\ 3$; the similarity in performance for the two small bases presumably reflects the fact that there are sometimes more than 3 relevant documents in the top 10, sometimes less. The difference is also, as before, significant. But unfortunately when we compare having limited relevance information with having none at all, there is either little or no performance gain. That is, as noticed above, while having an unrealistically large supply of relevance information, as in 'all', does improve performance compared with $QACW$, so $QACIW\ pred\ all\ >\ QACW$, having only a little relevance information is not guaranteed to contribute anything useful: $QACIW\ pred\ top\ 3\ =\ QACW$, and the same for rel in 10. The significance data in Table 6 agree for the top 3 case, though the difference for rel in 10 is significant. These results for $QACIW$ compared with $QACW$ hold regardless of request form, which appears surprising. But presumably, with only a few relevant documents, the frequency behaviour of terms in the document set as a whole is the dominant factor and the relevance information is not very discriminating.

5 Elaboration

The core model we have presented can be developed in various ways to take account of further sources of information. Exploiting these allows a more refined treatment of query-based indexing and matching and hence, potentially, better discrimination between wanted and unwanted documents. Some of the notions involved have indeed, in our own and other analogous tests, been shown to improve performance; the first we discuss, query expansion, is well established as very effective. However while the elaborations we consider in this section are all well motivated, their very complexity may make it more difficult to choose an appropriate instantiation of the general idea in question, and typically imply a requirement for some application-specific training. We consider questions of training in Section 8 later. But the practical difficulties of obtaining sufficiently varied training data in the form of a range of test collections mean that it is impossible yet to draw very form conclusions about some of the model elaborations we consider.

5.1 Query expansion from relevance information

It is evident that restricting query development just to providing term weights is not taking advantage of all the means of improving performance that are on offer. Specifically, restricting relevance feedback to reweighting is a very limited way of exploiting the information that retrieved and judged documents supply. Thus as Rocchio (Rocchio 1965) and Ide (Ide 1968) early saw, it is natural to consider terms in retrieved relevant documents as possible additions to a query for further searching (as also, perhaps, to eliminate existing query terms found not to be in relevant documents). While under the most abstract view of retrieval a query may be seen as consisting of all the terms in the file vocabulary, with positive or negative weights, we here consider only those terms which seem to have some connection with the query. Thus a case can be made for seeing terms not already in the query (as earlier), but nevertheless in documents marked as relevant to it, as potential query members.

It does not, however, follow that the candidate expansion terms, i.e. all those occurring in at least one relevant document, should simply be added to the query, even if they are then weighted using relevance information before searching. It may be better to select from the set, depending on the environment conditions that apply in the given application, for instance to avoid 'blowing up' an initial short query with an enormous number of terms that are candidates because documents are long. The constraints different conditions impose on selection are in fact not well understood (see Section 5.2 below). But under the assumption that selection is required, an obvious procedure for selection is to rank candidate terms and apply a cutoff to this ranking.

It might appear that the appropriate basis for ranking is the same as that for the ordinary term weighting using relevance information. But the question that is answered by a relevance weight ("How much evidence does the presence of this term provide for the relevance of this document?") is not actually the same as the question being asked here: "How much will adding this term to the request benefit the overall performance of the search formulation?". In particular, a very rare term, even though it is a strong indicator of relevance when it occurs, is not likely to have much overall effect.

A specific model for the overall effect is discussed in Robertson (1990). For each candidate expansion term, this considers the distributions of the scores for relevant and non-relevant documents, with the term in question present or absent. The model leads to a formula for a

'selection value' for term t_i, indicating the strength of its overall effect, which may be specified as follows (using the notation introduced in Section 2.6):

$$\text{selection value} = (p_i - \overline{p_i})w_i$$

In this formula, the ps are as defined as in Section 2.6, but the ws represent whatever weight will be applied to the term in question. The formula assumes, however, that the term weight is independent of the document — e.g. there is no TF or DL effect. We will therefore interpret this weight as the relevance weight RW or the collection frequency weight CFW, not as any of the combined weights.

In practice, $\overline{p_i}$ is generally very much smaller than p_i, and can in general safely be ignored. Moreover, if p_i is estimated by $\frac{r_i}{R}$, the denominator will generally be the same for all terms under inspection, and can therefore also be ignored for the purpose of ranking terms. These considerations lead to the definition of the following simplified selection value, the offer weight

$$OW = rRW \tag{16}$$

(ignoring subscripts).
Once selected, expansion terms are weighted in the usual way with RW.

Earlier experiments with the older collections reported in Sparck Jones and Webster (1980) explored various forms of expansion, but the tests were limited to the C1400I collection, and the conclusion that there appeared to be mileage in combining query expansion and relevance weighting was not followed up. The results for the T741000X collection in Table 5 cover a larger and more systematic range of experiments (though still for only one document file), and combine expansion with the more sophisticated type of frequency-based weighting described in the preceding section. Thus all of these experiments combine expansion with $QACIW$ weighting. We consider below the effects of different amounts of expansion. The tests here refer to moderate expansion, adapted (as seems reasonable) to the initial query request size. Thus we allowed (up to) 32 additional terms for the Long form requests, 24 for Medium, and 16 for Very short. These values, labelled 'exp 32' etc, are suggested by a range of experiments as being, if not optimal, at least reliably good for these requests; but the necessity to validate them in this fashion is a very good example of the need for training. For convenience we will refer to these in future as the *default* expansion set sizes, and also assume these sizes when characterising comparative performance results, unless otherwise stated.

The first set of comparisons is therefore between expanded runs, $QACIW + E$, and unexpanded $QACIW$ runs, for the corresponding prediction bases, namely top 3 and rel in 10. Apart from the Rec30 figures for the Very short requests, where there is no gain from expansion, both Rec30 and Doc30 show at least a Noticeable improvement and typically more, i.e. in most cases $QACIW + E$ *pred top* 3 $>$ $QACIW$ *pred top* 3 and $QACIW + E$ *pred rel in* 10 $>$ $QACIW$ *pred rel in* 10. This also holds for the significance tests, except for the one case of Rec30 for Long requests with top 3, where there is no difference even on the sign test.

Unfortunately, it is not possible with expansion to establish yardstick performance in the same style as for unexpanded requests. Properly, yardstick performance has to *include* the actual choice of expansion terms. But this is a serious combinatorial problem, essentially retro-engineering the perfect request at the level of term choice. We have therefore adopted a more limited approach, arguing by analogy with the earlier use of retrospective weighting on the assumption that the given request terms are the proper choice. Thus we assume that

the expanded query gives the proper choice of terms, so retrospective performance is about the proper weighting for these. Then for comparisons there are distinct sets of retrospective figures, one for each choice of expansion base, where each comparison is designed primarily, as before, to check the value of reweighting for each term set context. This is a somewhat more heuristic approach than in the previous case, but can still be practically useful.

Following this line of argument we can, at least semi-legitimately, compare retrospective performance with and without expansion. For the default degrees of expansion, and whether via top 3 or rel in 10, and for both Rec30 and Doc30, the expansion yardstick is at least Noticeably better, i.e., *QACIW + E retro top* 3 > *QACIW retro top* 3 and *QACIW + E retro rel in* 10 > *QACIW retro rel in* 10. These differences are also significant.

But the more relevant comparisons, for our Half collection, are between retrospective and predictive expanded query performance using, respectively, top 3 and rel in 10, and the default degrees of expansion. For both measures, retrospective performance is Materially better for the former, and at least Noticeably better for the latter except for the Long requests, i.e. *QACIW + E retro top* 3 >> *QACIW + E pred top* 3 and (except for Long) *QACIW + E retro rel in* 10 > *QACIW + E pred rel in* 10. Table 6 shows that the top 3 comparison differences are also statistically significant. However the differences for rel in 10 are not significant for Medium as well as Long requests.

As just described, query expansion is primarily an automatic strategy, requiring no more user involvement than judging documents for relevance. But of course expansion terms may be presented to the user, in *OW* order, for more participative query reformulation in searching. We consider this in more detail in the later Tasks section, Section 7.

5.2 Selective or massive expansion?

The question of whether expansion should proceed by selecting a few good terms, or by including every term that occurs in at least one relevant document, or even by including all the terms in the dictionary, is to some extent open. Potentially, every term provides some evidence (positive or negative) concerning the possible relevance of every document, and an appropriate weighting scheme would give each term an appropriate weight. If a term is strictly statistically neutral about relevance, the weighting scheme should give it zero weight, which is actually equivalent to excluding it, but would require no explicit exclusion step.

However, given that the expansion is based on a limited and fixed "sample" of relevant documents, estimating more parameters (i.e. weights) from the sample is likely to result in less accurate estimating of the total score. This argument has been described as the "curse of dimensionality" (van Rijsbergen 1979), and can be formalised mathematically (Robertson and Bovey 1982). Moreover, while the curse of dimensionality can be overcome by a suitable Bayesian prior distribution for the parameters, which would have the effect of damping the response of the estimate to small samples, it is not easy to define appropriate priors.

Empirically, some procedures for query expansion seem to be affected by the curse of dimensionality, in the sense that expanding beyond a certain point degrades performance. Other approaches appear not to be affected, at least to the extent that expansion with hundreds of terms has proved helpful. However the main results here, especially for so-called massive expansion, have been in TREC Programme routing experiments (TREC 1992-1997), where there have been rich requests and large relevance sets for training. Massive expansion was, moreover, found helpful only in earlier TREC evaluations; in later tests with shorter requests and different weighting methods it has not proved superior to more modest expansion

(Buckley, Singhal and Mitra 1996). In general, for the methods we apply, the evidence of past tests suggests that limitation is required, and this motivated the selective strategy described above.

At the same time, expansion does not seem to be affected by small changes in the size of the expansion set. The results in Table 5, for T741000X, illustrate predictive expansion for various set sizes for the different request forms, with rel in 10 as the expansion base. These show, for both Rec30 and Doc30, that increasing the expansion set for V requests from 16 to 24 has no effect, and that using 16,24 or 32 for M requests has no effect; with the L requests each step considered individually does not improve performance: only when the two extremes considered, expanding by 16 terms or 48, are compared, is there a Noticeable difference; however continuing to expand as far as 72 has no further value for the Long requests.

One suggestion, in the expansion context, is that the user's original query terms should always be given extra weight, and we have explored this to a limited extent using the following argument. We suppose that the use of a term in the original query is equivalent (as evidence) to its presence in a certain number of relevant documents. These may be supposed to be documents not in the collection, but previously known to the user. Instantiating this argument requires specific assumptions about numbers: we have assumed that the user knows about 20 relevant documents which are not in the collection, and each query term occurs in 19 of them (a fairly strong bias towards query terms). Table 5 compares predictive performance with and without query term emphasis, for the default expansion sets and using both top 3 and rel in 10 as expansion bases. This shows, however, that there is no gain from emphasising query terms.

5.3 Expansion without relevance information

Expansion using known relevance information has suggested a related search strategy using *assumed* relevance information, which is independent of user intervention (Evans and Lefferts 1995). Thus top-ranking documents from an initial pass are assumed relevant and terms from them are added to the query for the 'real' search. This is for obvious reasons likely to be much less effective than using real relevance information, and may be positively damaging for individual queries with poor initial searches. The utility of this strategy clearly depends on the characteristics of user needs, requests, and document files, but there is some evidence to suggest that the procedure can be beneficial when initial query quality is good. It has certainly become popular in TREC (Sparck Jones 1998c).

Since this *blind* strategy would be a practically convenient one, we explored it for the T741000X collection. The results labelled 'blind' in Table 5 illustrate performance when best matching documents are used for the default degree of expansion, using the 10 best for the Long and Medium form requests, but a more cautious 7 best for the Very short requests. Comparing expansion using either top 3 or rel in 10 with this blind expansion shows some difference in results for the two measures: for Rec30 performance using known relevant documents is sometimes the same as for blind expansion, and at best only Noticeably better; with Doc30 known relevant documents appear more effective. More particularly, expansion using top 3 is not necessarily better than blind expansion, but expansion by rel in 10 is at least Noticeably better than blind expansion, i.e. $QACIW + E\ pred\ rel\ in\ 10 > QACIW + E\ pred\ blind$. However this difference is only statistically significant for L and M requests, not V ones.

The second comparison is between no expansion and blind expansion. This also implies no

known relevance information for weighting, though of course it is possible to reweight query terms using the data from blind feedback. The comparisons are therefore with $QACIW$ + E *pred blind* against $QACW$, and against $QACIW$ *pred blind*. These show a gain with blind expansion compared with $QACW$ which is at least Noticeable for Long and Medium requests (i.e. $QACIW + E > QACW$), though there is no gain for Very short requests. The same applies to the comparison with and without expansion, i.e. $QACIW + E$ *pred blind* $>$ $QACIW$ *pred blind*, except for V. Not surprisingly, as the foregoing implies, reweighting alone using blind feedback gives no improvement, i.e. $QACIW$ *pred blind* $= QACW$. The statistical significance tests confirm this pattern.

Query expansion in general is a broad notion, and one that covers query development before as well as after any searching. It is assumed, probably rightly, that initial queries are typically rather sparse, and could therefore benefit from expansion before searching at all; and many methods of forming fuller queries have been tried, for example taking words or phrases from thesauri. Any expansion method from outside the probabilistic model could, in principle at least, be combined with the probabilistic model, in the sense that new terms could be weighted in the same way that original query terms are. One might however argue that such terms should be given lower weighting status than the original query terms. But this has to be done pragmatically, and hence in some fairly arbitrary way, because at present there is no formal justification in the probabilistic model for any specific way of doing it. There is the further point that the queries most in need of expansion (say 1-3 initial words) are also those for which there is least initial leverage for expansion. This in itself is is a strong argument for using relevance information, since this must provide more leverage.

It is also possible to apply feedback, using search output, in a more informal and less constrained way than in the model-based expansion we have described. Thus output might suggest thesaurus entry points to the user, or otherwise prompt the user to supply further terms. But these strategies could be handled within the model either by treating them as new initial query terms, to be given non-relevance weights or, if output has been assessed for relevance, by giving them relevance weights. Similar problems to the above, concerning the relative value of such new terms compared to the originals, exist in this case. Another open question would be: given relevance information, is it better to expand automatically or with user intervention? This question is further discussed under tasks in Section 7.

5.4 Term cooccurrences

In the Section 2 we noted that the assumption that terms occur independently, on which all of our model development so far has been based, is patently incorrect: given that topics are complex, terms expressing them can be expected to cooccur. We will get term dependencies, where dependency is statistically defined as co-incidence in units on the scale of whole documents. Thus in principle, the formulation of probabilities about documents being liked given single terms should be elaborated to cover probabilities about documents being liked given combinations of terms. It is certainly desirable that weighting schemes that assume independence should not be badly thrown if dependencies actually occur. However if taking account of dependencies implied doing without *any* independence assumptions, we would have to make separate estimates for *every* possible combination. This is normally impossible (even for original query terms sets, let alone expanded ones), because there are far too many such combinations and far too little external evidence on which to base such estimates.

A number of different ways of dealing with dependencies have been proposed. These

include:

- adopting term weighting schemes that are independence based but are robust or not sensitive to dependencies, e.g. van Rijsbergen (1977);
- applying document scoring functions at search time that reflect average dependencies when term weights are combined;
- including specific term pairs as if they were separate terms, and allowing those particular pairs' weights to differ from the normal sum-of-weights, e.g. Robertson and Bovey (1982), or in some other way allow for specific dependencies;
- using observed term dependencies in the collection to suggest query expansion terms before relevance feedback.

Unfortunately, experiments with indexing and searching exploiting these ideas have not delivered substantial performance improvements. The presumption must be that the information available about term dependencies is in general not strong enough to help with what is in fact a coarse task, especially where other factors such as the number of terms in the initial query may be more significant for performance.

At the same time, it may be observed that the independence assumptions in our model actually imply some dependence between terms. We can avoid the need to make any explicit estimates of probabilities for combinations of terms because these are implicit in the single term probabilities. Thus if we take two terms that are good for the query (i.e. both occur more in relevant than in non-relevant documents), then independence in each of these two sets implies co-dependence in the document collection treated as a whole. While this association (based on relevance to a particular query) cannot possibly explain all dependencies between terms, it may be enough to capture (on average) the most important ones. Given also Cooper's observation (Cooper 1995) that a generalisation of the independence model, known as *linked dependence*, leads to the same equations, it appears that these equations are pretty robust against distortions due to term dependencies.

Some further observations on independence are made in the Comparisons section, Section 9. One effect of dependence under an independence model may be to over-inflate the score of documents containing several of the terms. This may not be critical from the point of view of document ranking, but it may have other implications, e.g. concerning phrases (see below).

5.5 Term phrases

Though we have so far accepted that the general probabilistic model may be applied to any type of index term that is viewed as an integral unit, we have assumed that in practice terms will be simple single words or stems. However *compound terms*, or phrases, may also be used (Sparck Jones 1998a). As is common in text retrieval, *phrase* will be used to describe a compound term, generally a noun phrase made of contiguous words, even though this is an over-simplification of the proper notion of a phrase. Indeed as has been common in the SMART Project work (Buckley, Allan and Salton 1995), for example, phrases have been defined purely by co-location (adjacency, proximity) and selected on statistical grounds, thus indirectly capturing a significant linguistic relationship. Tight co-location of this sort

imposes strong constraints on term dependency, so phrases could prove more effective than the document-level term co-incidence just considered. However while, just as with the weaker dependency, a statistical basis might suggest that phrase information could be handled within the model, it does not follow that this could be finessed via the independence assumptions as in the earlier case. The very individual, close relationships between the words in a phrase may need to be explicitly acknowledged (Sparck Jones 1998a).

Phrases may be handled in a number of ways, which we may briefly summarise as follows. They may be defined at file time, or at least prior to any query formulation, manually or automatically, the latter (as noted) either through natural language processing or through the application of statistical association measures. If they are defined at file time, they may also be actually assigned to documents at that time, whether manually or automatically. Alternatively, the file-time indexing may be done with single words, and any phrases may be identified only at search time. In this case, in addition to or instead of pre-defined phrases, query-specific phrases may be identified: either manually (by the searcher), or by some automatic process on the query text. A phrase may be treated as an undecomposable unit; alternatively the phrase *and* its component single words may be regarded as indexable and/or searchable terms.

Clearly, there are many possible combinations of these variables. The following discussion covers only a few of them.

Undecomposable phrases

Pre-defined undecomposable phrases applied at file time (whether automatically or manually defined and whether automatically or manually assigned) may be dealt with in a very straightforward way in the probabilistic model. They can be treated as strictly equivalent to single words or stems, with weights dependent in exactly the same way on exactly the same data (which would also be collected in the same way).

Predefined undecomposable document phrases applied at search time may also be treated similarly. The only difference is that indexing/searching mechanisms must allow for searching on phrases, and also for the derivation of any required data (e.g. within-document frequency of the phrase). Modern inverted-file techniques, including full position information, can provide the basis for such mechanisms as long as syntactic checking is not required. The same comment would apply to undecomposable phrases introduced at search time: for example, if a user's specification of a phrase as such is to be taken as an instruction not to decompose it. Such a user construction (which might extend to any other possible user construct, such as term proximity – "within n words of", or "in the same sentence as") may be taken to define a new search term which should be treated exactly like any other. Again, the only constraint is that the system should be able to recover the necessary data for weighting.

Decomposable phrases

The situation is, however, rather more complex that the foregoing suggests as soon as compound terms are treated, as is nowadays common practice, as automatically decomposable. Thus in the past in conventional systems with Boolean searching, complex descriptors or terms were viewed as autonomous but might be explicitly replaced in further searches by 'reduced' versions of the terms, in response to initial matching failures. Within the modern postcoordinate framework, it is usual to include both compound terms and their constituent

elements - e.g. both a word pair and the member single words - in the final search query. The presumption is that preference will normally be given to matches on the phrase because it will have a higher weight than either member, so a document containing the phrase will be ranked higher than one with just a member term, or than one containing both terms but not in the phrasal relationship. This preference is simply an automatic consequence of the nature of the data used for weighting.

Using *both* phrases and members as query elements is also a natural strategy for full text searching for another reason, since they may occur independently at different text locations and matching on both can be taken as reinforcing the topic relationship between the query and the document. This is obviously appropriate when user requests are themselves extended texts, but 'unpacking' any query to allow for multiple matches against an extended document is intuitively reasonable. Indeed it may also be argued that counting both phrase and member matches for the same text location is justifiable as a way of marking shared topicality, quite apart from whether it happens in practice because of the way matching is operationally implemented. Furthermore, the restriction to undecomposable phrases tends to reduce the power of best-match systems to discover less-than-perfect matches, which is one of their great advantages.

There is, however, a substantial problem with the interpretation of the probabilistic model in the context of decomposable phrases. We may reasonably guess that if both phrase and constituent words are given weights in the usual fashion, using one of the formulae discussed in this paper, then any document containing the phrase is likely to be scored too highly. This is a particular example of an extreme form of dependence between indexing units: the presence of the phrase *entails* the presence of the single words. In theory, the weight of the phrase should reflect not the increased odds of relevance implied by its presence as compared to its absence, as a whole unit, but the increased odds compared to the presence of its component words. Furthermore, for consistency one should probably *reduce* the scores of those documents which contain the component words but not the phrase, e.g. by giving a small negative weight to the logical conjunction of the component words. This is particularly appropriate given that combinations of words are generally over-scored in the independence model, as indicated in Section 5.4.

Experiments with schemes which give a boost to phrases over the weights of the component words have generally shown only slight benefit. Our experiences here mirror those of other TREC participants (Mitra et al. 1997). The analysis of TREC in Sparck Jones (1998c) shows some leaning towards the use of phrases (variously defined) in TREC. This has become part of the 'default' automatic strategy that also includes blind feedback and the combination of document and passage information either in some form of global/local strategy for scoring matches or for bounding sources of expansion terms (see Section 5.6). The principle in using phrases seems to be that as these may be of value, especially in conjunction with other devices, and do no harm, it is sensible to have them. The evidence in their favour is not, however, strong enough to call for a range of phrase tests here even with some adhoc mode of treating weights, since the range of system parameters to explore would be large (e.g. using syntactic analysis or not, using a fixed phrase vocabulary or defining phrases at search time, using strict adjacency or a proximity window, etc.)

5.6 Document levels: passages

Just as providing a comprehensive treatment of terms may be more complicated than in our initial development of the model, so there may also be good grounds for adopting a more sophisticated approach to documents than hitherto. Thus it may be helpful to consider the evidence that document structure, more particularly information at the level of subdocuments or *passages* within documents, provides for probability of relevance.

A variety of approaches to the use of passage-level information have been proposed. Some of the possibilities are:

1. To divide the original documents up into discrete passages, either purely arbitrarily or following some explicit structure indications like paragraphs, section headings or mark-up tags. These passages may then be taken as documents in their own right, so the matching process with the query then deals with passages only.

2. Not to make a discrete division, but to attempt a match on any possible passage in the document (including the entire document), and choose the best-matching passage to retrieve.

3. To divide documents up into discrete passages, but to combine passage-level information with document-level information in the scoring process: a document's matching score with a query may then be a function of the score of the entire document and that of, say, the best-matching passage.

Other methods can also be devised. Thus it would be possible to have a two-level matching scheme involving first selecting whole documents and then selecting passages within them. Passages may also be defined by word length (window size) rather than structural criteria, perhaps with overlap to ensure natural topic units are de facto captured.

These methods are considered here only from the point of view of retrieval logic, and not in the context of output presentation to the user. Thus in the first case, whether passage-to-document links are completely thrown away or are retained so users can invoke the source documents for output passages is a management or human factors issue outside the retrieval logic. Equally, in the second case, whether the best passage is a good presentation unit is a separate matter. In the third case the presumption is that output will be at document level, in the conventional way, but as information about passage value is available it could be used to lead the user to the best passage within documents.

Passage retrieval methods are sometimes classified as "local only" (i.e. using only passage-level information), or "global/local" (combining passage-level with document-level). The first method above is essentially local-only; the second is logically local-only, with the proviso that the local includes the global; the third is global/local. However there are also other perspectives on the use of passages, namely whether passages are uniformly defined for all documents (or all documents of a given type), or vary for individual documents: this reflects a difference between passage definitions that are *static* or *dynamic*. In the first local-only method passages are static. In the second, in contrast, the best-matching passage in one document may be of a different kind (structural unit, size) from another, so the method is dynamic.

Of the methods suggested, the first and second (the two local-only methods) could be used with the weighting and scoring functions proposed in this paper, without modification. The

first, however, requires the determination of suitable passage boundaries, for which there is no obvious model rationale. It is also less flexible than the second (if passages discovered by the first method are good, they will also be discovered by the second method, but not necessarily vice versa). But the second method has the disadvantage is that it is more expensive: a great many more matching scores have to be calculated. It will also only work if the weighting and scoring functions take due account of document length, so that the comparison of the score of (say) the whole document with one of its constituent passages is valid. However the document length weighting function we gave in section 4 earlier naturally serves this purpose.

The reason that the (global/local) method is not immediately suitable for the probabilistic approach suggested in this paper is that it would require a more complex model. For example, one might identify relevance of the passage and relevance of the document as separate variables, with separate probabilities. No such model has yet been developed.

The second local method has been tried with the weighting functions defined in this paper. There are sometimes performance improvements, but not substantial ones. While this could be due to some failure to interpret the notion of passage properly, a more likely explanation is that given for the lack of improvement with phrases. This is that the model already takes account, sufficiently for the coarse retrieval task, of the correlation between the treatment of a topic and the unit of text that treats it. Thus the model's weighting function, which already takes account of document length as putatively correlated with topic coherence, seems to reduce the need for passage-level information. However though the model appears robust in the sense of not being oversensitive to fine data features, it has not been tested with document sets that would allow systematic exploration of the properties and roles of passages, or of the combination of retrieval mechanisms exploiting passages and user interests in passages.

Some tests on the use of passages are reported in Table 5. These are essentially of the second type listed above, but with certain compromises for efficiency reasons. Each document is divided into atomic units which are paragraphs (often quite short, only a sentence or two, in this material). A passage is an arbitrary window on these paragraphs, which can in principle start anywhere and be of any length; the score of a document is that of the best matching passage within it. Since one of the passages under this definition is the entire document, this competes with the shorter passages.

The efficiency compromises are as follows. An initial search is performed at document level only, and the top 10,000 documents are selected. All passage work is done within this set. Passage windows start anywhere, but are limited to lengths of 1–10 paragraphs. This rule might possibly exclude the entire document, but that exclusion is overridden, so that the whole document is always considered; however, longer intermediate passages are not.

The results are somewhat variable. Comparing first retrieval without expansion, using $QACIW$ and weighting via rel in 10, for both Rec30 and Doc30 measures there is no Noticeable difference. The same applies with query expansion, $QACIW + E$, using the default expansion sets. Recent TREC work (cf (Sparck Jones 1998c)) suggests that using passages round matching query terms to bound the sources for query expansion terms seems to be useful. We have not tried this yet.

5.7 Final note

Overall, the conclusion to be drawn in relation to the various elaborations of the core model that we have examined are that where these refer to refinements of primary indexing or searching beyond that supplied by the core term weighting, but without reference to relevance

information, there is little performance gain. But this must also be because most of the devices in question do little to change the character of the query. Thus where relevance information is exploited, and is applied to make substantial changes to queries, as in relevance expansion, the effects on performance are much more significant. From the model point of view we get a response to large, important data change, as opposed resilience to small, unimportant data variation.

6 Environment conditions

6.1 Document properties

We have already noted that the test collections summarised in Table 2 and used for the experiments illustrated so far are very different in their characteristics, especially in relation to document length. Thus the test collections used in the TREC Programme led to developments of the model to deal with long documents, and in the model instantiation to the incorporation of the constant K, to be set empirically in relation to typical document length in Section 4.6.

There is some presumption in this that the variation in document length within a file is not too great to render the verbosity hypothesis untenable or make this simple strategy unviable. Alternatively, to cover the case where greater document length is in fact typically associated with content heterogeneity, we could also apply the global/local strategy discussed in the last section.

However variation between documents is important not so much in its own right as through its impact on file composition. The TREC collections are more varied in other document properties than just text length and topic homogeneity, i.e. they include text documents of very different genres or types, such as news, patents, government publications, technical abstracts, etc. In general the TREC Programme tests, as illustrated by the City papers in TREC (1992-1997) and by the surveys in Sparck Jones (1998b, 1998c), suggest that the probabilistic approach is quite robust under considerable collection heterogeneity, though we may guess that performance could be better with one subcollection than another. It is also the case that the model is in principle applicable to documents of far other kinds than those covered by TREC, including those with little running text, great informality, distinctive sublanguages etc. Systematic testing has not, however been carried out to determine either characteristic performance for really different types of material, or the impact of mixing these in a single file: this is a potential problem for any application of the model to the World Wide Web, for example. We may note, however, that in the Very Large Collection track experiment in TREC-6 (Hawking and Thistlethwaite 1998), scaling up to a much larger collection (some 7.5M documents) also involved handling a larger range of material, including USENET News.

It is however appropriate to note here that the model has been applied, albeit on a very small scale from the point of view of file size, to spoken documents, in the Cambridge VMR Project (Sparck Jones et al. 1996, Jones et al. 1996). The primary purpose of this work was to study the application of speech recognition techniques in the retrieval task, and to examine the impact of imperfect word recognition on system performance and retrieval strategy requirements. The test collection was too small for properly grounded retrieval tests, and thus only suggests, rather than demonstrates, that when the model is applied to noisy data with the characteristics of spoken documents the same performance relationships between the model interpretations instantiated as CFW and CW respectively appear to hold as in the ordinary text case, and that probabilistic weighting performance with CW could be expected to be superior to the coordination baseline.

6.2 Request properties

Even for the routine adhoc retrieval task with which we have been concerned so far, there may be great variation in requests as expressions of user needs, and hence in the term composition of initial queries. This variation includes not only e.g. differences in concept generality and other content properties that are not directly acessible though they may be inferred in the

model from term incidence frequency; it also covers differences in care and elaboration which are also not directly accessible but may be inferred from the number of terms provided, and are indeed exploited for retrieval in the basic postcoordinate approach.

In general, very short initial queries are associated with poor performance, and provide little leverage for developing better final queries (as noted in connection with query expansion). Encouraging end-users to provide good starting requests (and hence queries), for example for searching the World Wide Web, is further examined under Open Issues later. However we note here that it is one of the major strengths of the probabilistic model that the methods of term weighting drawn from it lead to relative performance improvements even where absolute performance, through poor queries, is not impressive. This is illustrated by an overall comparison between performance for Long, Medium and Very short requests as shown in Table 5, Thus performance for the V requests is generally considerably below that for the L version, but the same strategies as improve performance for L and M improve it for V, except that the first two gain much more from expansion. The TREC-5 and -6 tests also supply some confirming data here. The former offers 'Short' (in fact Description field only) versus 'Long' (full topic) comparisons for rather difficult requests, the latter 'Very short' (Title field only), 'Short' and 'Long'. Unfortunately Title terms were not necessarily included in the Description field, and for TREC-6 in particular the Title had it own considered character; the comparisons illustrated in Sparck Jones (1998b, 1998c) thus show 'Very short' queries performing better than 'Short'. But these comparisons also, more importantly, show City performance with automatic searching as among the best for any of the request forms, though the absolute performance levels vary.

6.3 Languages

All the results illustrated so far have been for English. Dealing with searching over multilanguage files (and so involving query translation) has not been attempted because suitable test data is not available; whether it presents special problems in model terms is not clear, since this may depend on the strategy used: e.g. if queries are exploded because individual input words are replaced by all their possible equivalents in each other language.

However experiments, though not yet very extensive ones, show that the model carries over to other languages. It would indeed be very surprising, given its generality, if it did not. Thus for the TREC Chinese material discussed in Smeaton and Wilkinson (1997), the test results obtained by City are useful as initial evidence that the model interpretations not only carry over to other languages, but also to ones where the nature of basic terms is somewhat different from that of English words.

7 Tasks

7.1 Interactive searching

We have already considered the natural extension of adhoc searching to iteration, especially in the interactive session mode. Thus we have interpreted the model to exploit user relevance judgements to reweight existing query terms and to find candidate expansion terms, which may be offered to the user for explicit consideration and not just be simply automatically added to the query. It is evident that, as a practical implementation matter, it is important to encourage the user to provide the relevance assessments on which future performance depends, and this is one aspect of the general question of interface design which is an important issue in its own right. There are, however, also more fundamental questions bearing on our approach: about the reliability and authority of user-offered and system-gathered information respectively; and about whether interactive searching differs in subtle ways, as a task, from the old-style one-off search with which we started, so that from the model point of view interactive searching has its own requirements and presents its own problems.

Thus in relation to task status, it may be that the user's beliefs about the nature of documents in the file and about the file as a whole lead him to make term choices that are in fact poor, however rational they appear to him from a topic characterisation point of view. Ultimately the user has to be king, but the question is whether the system can provide sound guidance for his query development. Thus by offering candidate expansion terms specifically in OW order rather than simply any independently-related terms, for example ones from a manual thesaurus, the system can help the user to choose really effective rather than apparently useful new terms. (So the interface design has to make sure this system behaviour is persuasive as well as sound.)

In relation to task type, it may be the case not only that one-off retrieval, i.e. search for a given need, becomes more complex as it subsumes an iterative sequence of searches within a session: with the development of the query through successive forms, especially if this is based on inspection of retrieved dcouments, the need itself may be changing, so it is less reasonable to think of iteration as convergence on the best query expression of the original need. But more importantly, even if the latter can be treated with the model in the same way as the routing task considered below (though perhaps less effectively in practice through having less relevance data), in interactive sessions the task may not be retrieval in the straightforward sense we have hitherto assumed. It may become other tasks, like browsing or item location, or become a complicated structure of interdependent tasks which should not, or cannot, be decomposed into an orderly series of distinct different tasks.

The Okapi system using the probabilistic model described here is an operational as well as experimental one, and has been used in a series of tests with real users with real information needs (Okapi 1997). This is a large area of research, quite distinct from the kind of laboratory experiments used for the present paper and in TREC. It tends to concentrate on issues concerned with interface design and other aspects of HCI, and on user perceptions of system functionality. A central concern, for example, is with how users perceive the relevance feedback function, and how they can integrate its use into their retrieval strategies and tactics; another has to do with user selection of system-offered expansion terms, for example after relevance feedback. To a limited extent, such questions can also be asked in a laboratory environment, and the interactive track of TREC attempts this; but investigations in operational contexts are essential. These issues are discussed extensively in Okapi (1997).

7.2 Text extraction

The possibility of operating below the document level, with passages, as discussed in Section 5 above, has several implications in the context of different tasks. In interactive searching, for example, even if passage retrieval is not essentially more effective than whole-document retrieval, presenting the user with the best-matching passage may have considerable benefits over giving him/her the whole undifferentiated document. Even better, summarising the document by extracting a small number of key passages might be a major advance for some users: see for example Salton et al. (1997). Documents retrieved in response to a request might be summarised as a group by a similar method. However evaluating extracted passages as summaries in their own right is a major challenge, and establishing their value within an interactive search context is also tough, though a current DARPA evaluation is doing this in a limited way (Hand 1997), by checking whether summaries are reliable as bases for relevance assessment. Seeking to apply the current model to passage extraction is an area for further work: it presents a challenge because, though extraction may use the same facts about term occurrences and methods of exploiting these as retrieval does, it is not at all clear what relevance means here, and thus what the model-based justification for the extraction techniques is.

7.3 Document summarisation

A set of terms taken from a document text may be deemed a minimal summary, and while single words may be over minimalist, a set of phrases (however defined) may be surprisingly informative as to the topic of the document. The criteria for choosing terms for this purpose should be aimed at representing the content of an already selected item rather than in selecting if from among similar ones. But there is no reason in principle why methods exploiting the probabilistic model should not be applied to key term choice for 'mini-summaries'. We have not explored this, but work in the general area has begun (Mani and Maybury 1997) and it is clearly a line for development of the model, subject to the same considerations about its reliance on relevance as hold for extraction.

7.4 Routing and filtering

There are, however, other tasks still within the retrieval area to which the model is also applicable and to which, having the advantage that experiments can be comparatively easily performed, it has already been applied.

One such task is document routing or *filtering*, earlier often called selective dissemination of information.

The general task specification for filtering differs distinctively from the adhoc case. The most important characteristic of adhoc retrieval is that there all the documents in the file are considered and ordered in relation to the query. Thus, although each document is separately matched, performance is affected by the composition of the file at search time. There is also, normally, no opportunity to learn from past performance for the individual request beyond the rather narrow limits of the iterative session. This in particular is likely to mean that the number of relevance decisions on which a better new search can be based is small.

With filtering, in contrast, there is a stream of documents and, for each incoming document, the system has to make a separate allocation decision as to whether to assign it or not to each member of the user profile set. The decision is a critical yes/no decision: there is no

ordering of a set of documents, or chance for the user to stop at some point in the ranking because they have already found what they want. However as against this, there is usually far more opportunity for the system to learn and improve performance than in the adhoc case. Thus a series of relevance judgements can be built up over time with little user effort, which has two advantages. One is that the absolute amount of relevance information available to modify queries can be large. The other is that as the information is acquired over time, the query can be gradually modified to reflect any slow change in the user's need as implicit in his judgements.

It is evident that filtering is a natural field for the probabilistic model, and that its specific interpretation to use relevance information is well suited to this task. The TREC evaluations so far have, however, allowed only one aspect of filtering to be studied. This is the value of large amounts of relevance information for query training. Thus in the so-called TREC routing task, searching has been with old initial queries on new documents, as in the filtering case, but there has not been a stream of documents to be treated one at a time. Evaluation has been by conventional methods, as if the new collection was being used in adhoc mode. This is clearly unsatisfactory, and after initial studies focussed on suitable performance measures (Lewis 1997), work is underway on tests closer to true filtering, with document streaming. However some of the results of the TREC experiments we have presented can be taken as showing how the probabilistic model profits from rich training data. Thus in Table 5, if we compare $QACIW + E$ *pred rel in* 10 (something like the best reasonably achievable with relevance information) with $QACW$ (the best without), the former is Strikingly better than the latter on Long and Medium requests, though the difference is only Noticeably so on V.

In fact, under these artificial test conditions, we can do a lot better, by performing an iterative optimisation of the query formulation on the basis of the training set of known relevant documents. No such results are reported here, as there are many variables to be investigated, but such methods have proved their power in successive TRECs (see City papers in (TREC 1992-1997)). TREC routing experiments by various teams have further shown that with rich training data more query expansion than in the adhoc mode can be advantageous, though it is still possible to overdo the expansion. Iterative optimisation methods have also been shown to work well under a measure of filtering performance based on yes/no retrieval rather than ranking (Walker et al. 1998), albeit not in a fully 'streamed' situation.

7.5 Absolute scores

One characteristic of the probabilistic model is brought into question by the filtering experiments. At several stages of the argument in Section 2, appeal was made to the idea that the purpose of the scoring method is to generate a ranking of the documents; thus any convenient order-preserving transformation of the scoring method is permissible. For filtering, one would want the scoring method to give an absolute value for the probability of relevance. This point is reinforced by the measures of filtering effectiveness being introduced in TREC (Lewis 1997) – these are utility measures, which could be used directly to define thresholds on the probability of relevance. There may also be other circumstances in which an absolute score is desirable, for example if we wish to give the user an explicit estimate of the probability of relevance for each document.

It it not at all obvious how to recover an absolute score from the probabilistic model. If we were to retrace our footsteps and avoid the order-preserving transformations, we would then have to obtain estimates for various other quantities, specifically $P(R)$, the unconditional

probability of relevance to a given query of a document taken at random from the collection. This probability typically varies very substantially between queries, and we typically have no information on which to base an estimate.

A possible alternative would be to do some post-hoc calibration on one of the scoring functions proposed earlier. It may be feasible to use regression-based methods, such as those discussed in Sections 8 and 9 below, for such calibration.

7.6 Indexing and category assignment

The idea of indexing documents, in the traditional sense of assigning index terms of whatever kind, has not figured strongly in this paper. Essentially the basic probabilistic model assumes that the document descriptions are in some sense fixed, and that the object of the model was to improve the representation of the queries only. While such matters as the use of phrases (discussed in Section 5 above) might be dealt with at index time, this was regarded as a decision outside the framework of the probabilistic model. By and large the mechanisms suggested by or in the context of the model, including (e.g.) query expansion, are designed to improve the queries alone. Furthermore, all the indexing elements that have been considered so far have been assumed to be easily identifiable units in the natural language of the documents (words, stems, perhaps phrases).

It should also be pointed out that nothing in the foregoing precludes the use of other kinds of assigned or extracted indexing. Index terms taken from a thesaurus, classification or category codes, even author names, citations, or publication details, might be *used* by a probabilistic searching model, whether as constituents of the initial request or as grist for query expansion, though they might have to have been provided by some external process. The only constraint is that these descriptors could be treated within the coordination framework, regardless of whatever structural relations might in principle be associated with them.

However, the question then naturally follows as to whether the type of probabilistic model we have presented could be used more directly in the categorial indexing process. We may imagine a number of different scenarios in which this question might arise: e.g.

- We have a number of predefined categories or index terms, and perhaps a number of examples of assignment of these to documents by human indexers, and wish to automate the assignment of these to future documents.
- We have nothing predefined, but wish to identify a suitable set of categories for a collection of documents.

Both of these situations seem to be candidates for probabilistic approaches, and both have been subject to a range of broadly statistical methods in the past, e.g. by Biebricher et al. (1988). However, there is one substantial difference between most such work and the model discussed in this paper: that is, it is rare to appeal directly to a notion of relevance.

As an example, consider the first of the two situations mentioned. It is quite possible to devise a formal probabilistic model to describe this situation, which would involve such notions as

$$P(\text{category assigned}|\text{NL word occurs})$$

This model would allow the system to learn from the training set about how indexers assign categories, without reference to queries or relevance. Similarly, much work on document

clustering tries to detect which documents go naturally together, in terms of their given characteristics, but without reference to the queries to which they might be relevant.

The question of how to apply our notion of probability of relevance to indexing problems is an open one. One approach is the probabilistic indexing idea of Maron and Kuhns (1960), but this involves assuming the queries remain fixed. An attempt to unify the Maron/Kuhns model with the Robertson/Sparck Jones model is given in Robertson, Maron and Cooper (1982), but is substantially more complex. These ideas are discussed a little further in Section 8, in relation to the different kinds of learning or training that might be embedded in a system.

8 Training

Many modern approaches to retrieval involve some notion of training, and much current research is focussed on training in one way or another. The scale of application runs from the most modest system *tuning* at one end, to fullblown system *learning* at the other. Discovering appropriate values for critical system parameters, like k_1 or b, from training collections illustrates tuning. Automatic relevance feedback mechanisms illustrate learning: the system is designed to learn the characteristics of the documents that a user judges relevant to a query, in order to discover more of them. Iterative methods for determining internal parameter settings during system operation, for example to choose the number of expansion terms for a query, or the best passage size for a document, might be said to fall into intermediate positions on the training scale.

We may also analyse training opportunities and requirements according to how general or specific the learnt characteristic is: a value of k_1 relates (we hope) to a large class of queries and documents; a relevance feedback weight, on the other hand, relates only to the current query/need. It is desirable to train for general characteristics because retrieval systems are intended to support many searches; but as the relevance feedback case emphasises, specific training puts icing on the cake. More particularly we have to consider to what training is being applied, for what purpose, and with what data. Thus we may train to determine the distinctive characteristics of specific terms, documents, requests, or users (across searches); or to determine the characteristics of classes of terms, documents, queries or users, to a lesser or greater degree of generality; or any combination of these individuals / populations. Training may be for adhoc retrieval or filtering, for instance, imposing different requirements. Each application of training, with respect to object and objective, then determines what particular training data has to be collected and, less or more particularly, the form of training procedure.

Training for global parameter settings, like those discussed in Section 4, requires collections of documents, queries and relevance judgements that are complete or very extensive, though it may be difficult to in practice to ensure that these collections have the same properties as those to which the trained system will be applied. Indeed it may be difficult in practice simply to get enough data for adequate training. However in other cases, like relevance feedback in adhoc searching, we cannot expect much training data in principle (though in filtering rather more might be accumulated), so robust training methods are mandatory.

It is evident that, as the core probabilistic model is extended to include more types, or more complex types, of information, as well being applied in a range of different data and task environments, the requirement for effective training increases. However the model meets certain training requirements very directly: e.g. it more-or-less *instructs* us as to how to do relevance feedback. It also tends to offer other opportunities, though somewhat less directly, as described below.

8.1 Learning about the information need

Because user relevance judgements provide evidence about the user's real information need (or at least their perception of it), relevance feedback is a potentially powerful source of evidence from which a system may learn. Although one might use it to learn about such things as classes of users, or terms, or documents, or about this individual user's searching behaviour (which might apply in other searches), the probabilistic model presented in this paper specifically invites us to learn about this particular query / information need / search

session. The 'probability of relevance' might be assumed to have specific meaning in this specific context; but what meaning it might have if abstracted in any way from this particular context is not so clear.

Given a context in which 'probability of relevance' has meaning, relevance judgements are very obviously direct evidence about this probability. Some level of abstraction is still necessary: for example, we do not need to know anything about the probability of relevance of the documents that have been judged already, only about those that have not yet been assessed. This level of abstraction is provided by the identification of terms as carriers of probability. Thus the probabilistic model is quite specific about what might be learnt from relevance judgements about the terms in question. (Notice that this is not about these terms in general, but about their relation to the specific need.)

8.2 Learning about documents

The probabilistic model may be extended in principle to learning about individual documents: that is, to learning to what kind of needs this document may be relevant (Maron and Kuhns 1960, Robertson, Maron and Cooper 1982). This is actually a simple dual of the model presented in this paper: using the same kind of abstraction (via terms), we learn about the relation of a specific document to other queries which may contain the same terms.

However, this idea is difficult to apply in practice because, among other reasons, it requires the collection of vast amounts of data. We could only gain significant information about a specific document after that particular item had been seen and judged by a number of users. None of the existing test collections satisfies this requirement. This is the reason why we have not presented the substantially more complex version of the probabilistic model given in Robertson, Maron and Cooper (1982) in this paper. It may however be suggested that it is now plausible to envisage the collection of such data, in some environments at least: thus it may be possible to obtain data about multiple inspections and assessments of the same document from World Wide Web search records.

8.3 Learning about terms

As indicated, learning about either queries or documents requires abstraction via terms. However, the terms themselves are not treated abstractly – they are firmly rooted in use in individual queries or documents respectively. We might, by contrast, want to learn the characteristics of individual terms, as they occur in any documents or queries. Or we might want to move to a more abstract level still, and learn about general characteristics of terms and how much they might tell us about relevance in specific cases. An instance is the collection frequency of the term, whose importance we have already discovered.

The probabilistic model presented here has nothing obvious to say about specific terms outside of the context provide by their use in specific queries. In particular, the model embodies a strong view of terms from a linguistic point of view, a view of a Wittgensteinian sort but with an interesting twist. A word used in a query has its meaning defined by its use in that context; however the context is represented not just by the individual linguistic context of the query statement, but by the larger linguistic context of the request's relevant documents. The analogy is with the set of times a word is used in, say, a dialogue between two people. This has the implication that whether a word has a similar meaning in another context is not easily established, and analogously in the opposite way for different meanings

in different contexts. The probabilistic model sits happily with an abstract view of term meaning that implies, for instance, that dictionary-based lexical operations for, e.g. sense selection are inappropriate and hence unnecessary.

However, extensions of the model may indeed make connections between uses of the same term in different queries. At the level of general characteristics of terms, we have seen how such parameters (specifically collection frequency) may emerge directly from the model. There are however many other potentially useful characteristics which do not do so, some of which have already been discovered to be useful predictors of relevance: for example whether a word is a proper name. We might attempt to build models which include such characteristics as internal parameters. Alternatively, we may treat them as external independent variables and build models which may be tuned to them. The approach in this paper to within-document term frequency and document length might be described as somewhere in the middle of the spectrum: the model does include these parameters as internal variables, but in an underspecified fashion, and leaves scope for tuning via the various constants.

8.4 Regression

One statistical approach to the problem of extracting general patterns from a training set (which might then be applied in other situations) is the technique of regression analysis. Given a variable we want to predict (relevance), and certain other quantitative variables which we believe may contribute to the prediction, regression analysis of a training set may tell us the extent of each variable's contribution. We have to assume in advance a particular form or equation, but estimates of the specific parameters in the equation are derived from the regression analysis.

Conventional linear regression methods may be applied within the framework of the probabilistic model (Fuhr and Buckley 1991) , to both the indexing and the searching processes. However, given that the variable to be predicted is binary (that is, we wish to assign a probability to it), logistic regression methods are more appropriate (Robertson and Bovey 1982, Cooper, Chen and Gey 1994). This latter approach integrates better with the probabilistic model. The matter is discussed further in the Comparisons section, 9, below.

8.5 Data exploration

A somewhat less formal procedure than regression is to extract data from a training set and inspect it in various ways (for example by plotting graphs), to discover useful relations which may then be incorporated into models. An example of such a method was used by Singhal, Buckley and Mitra (1996) to refine the document length normalisation methods used in the SMART system. The availability of large quantities of training material (specifically from TREC) is likely to encourage further instances of such exploration.

8.6 Some pitfalls of training

It is clear from all the discussion above, that data-driven training (in whatever form) is potentially an extremely powerful mechanism for retrieval purposes. Many of the improvements in system performance that we have reported in this paper owe a great deal to training, as well as to the development of formal models. The same applies to large areas of current information retrieval research as represented by the TREC Programme tests.

However there are also some significant general problems that apply to all training-based methods of investigation. Essentially, we train systems to do past tasks; and we then apply the trained systems to new tasks. The power of training depends on the lessons learnt from the past applying to the future, and there are all too many reasons why this might not occur.

If, for example, we put a lot of effort into discovering the best method of retrieval for a certain set of TREC documents and queries, there is no guarantee that this will be the best method for future TREC material, let alone for anything else. Statistically speaking, we may "overfit" the data we have, so that the performance of our chosen method may depend on accidental characteristics of the training set. Qualitatively speaking, we may tune our system to particular types of queries or documents (as discussed in Section 6), or even to a particular set of instances.

One uncomfortable feature of these problems is that the more effort we put into training, the more we risk running into them. Thus for example an iterative method of optimising a query formulation for a filtering query, over a large search space, is almost guaranteed to discover some accidental characteristics which work for that data set, and probably for no others. We can illustrate this problem fairly graphically: suppose we have such a filtering query, searching long text documents, with a training set of known relevant and non-relevant items. Most long text documents contain typographical errors; many typographical errors occur only once in a corpus. If our iterative method homes in on a query consisting entirely of the typographical errors that occur in the training set relevant documents, it may perform perfectly on the training set, but hopelessly on any new documents.

We do not have general solutions to this general problem of training. All experimental results in information retrieval must be qualified by an awareness of the problem, and indeed participants in TREC, where the volumes of data available for training are much larger than before, have identified overfitting as an explanation for sometimes disappointing results.

9 Comparisons

The probabilistic view of information retrieval has inspired a number of very different approaches, models, methods and techniques. It is also true that many of the specific methods discussed in this paper have been used in the context of other, non-probabilistic (or not explicitly probabilistic) approaches. Many comparisons could be made, at the level of theories, models, techniques, experimental results, or whatever, between the ideas discussed here and those reported by other researchers. In this section, we make a small selection of such comparisons, concentrating on some major alternative or complementary views, and on ideas which may shed light on the foregoing discussions. (For further background see Sparck Jones and Willett (1997), and also the recent review of TREC in Sparck Jones (1998c).)

9.1 The vector space model

By far the best-developed non-probabilistic view of IR is the vector space model (VSM), most famously embodied in the SMART system (Salton 1975, Salton and McGill 1983). In some respects the basic logic of the VSM is common to many other approaches, including our own: see the discussion of properties (attributes) in Section 2. It is also true to say that the VSM is hospitable to other theories, and indeed there are implementations of probabilistic ideas within the VSM. However, the point of departure for the VSM is that the attributes are to be regarded as the axes of a space, and that the required measure of association (e.g. between documents and queries, what we have in this paper described as the matching score) should be a distance measure in this space. In principle at least, this motivation is very different from the 'probability of relevance' motivation which informs the present paper.

In practice the difference has become somewhat blurred. Each approach has borrowed ideas from the other, and to some extent the original motivations have become disguised by the process. Two examples may be given. The idea of relevance feedback originated in the context of the VSM (Rocchio 1965), but also fits very well into the probabilistic approach, as has been seen. Second, the experimental success of the form of document length normalization described in Section 4 inspired the SMART system researchers to rethink their own document length normalization (Singhal, Buckley and Mitra 1996).

This mutual learning is reflected in the results of successive rounds of TREC. Typically SMART, Okapi and some of the other systems discussed below are among the best-performing systems with relatively little to choose between them (at least compared to the range of performances represented). It may be argued that the performance differences that do appear have more to do with choices of the device set used, and detailed matters of implementation, than with foundational differences of approach.

9.2 Probabilistic indexing and a unified model

The first explicitly probabilistic model in IR was due to Maron and Kuhns (Maron and Kuhns 1960, Robertson, Maron and Cooper 1982). While it is concerned with probability of relevance, it starts from the opposite end from us: user queries are assumed fixed, but document indexing requires optimization. No real experiments have ever been done with this model.

An attempt has been made to unify the Maron/Kuhns model with the Robertson/Sparck Jones model (Robertson, Maron and Cooper 1982). As indicated in Section 8, this unification

suggests the possibility of using relevance feedback both locally (for the immediate query) and globally (to modify the document indexing for subsequent queries). Again, this model has not been tested experimentally, although some other techniques directed at the same end have been (see the discussion of Fuhr's work below).

9.3 Dependency

Following the original Robertson/Sparck Jones model (with its assumptions of independence of terms), a substantial amount of work was done (e.g. by van Rijsbergen and colleagues, see Harper and van Rijsbergen (1978)) on formal models which made some attempt to avoid or relax such assumptions.

These models were tested to some extent at the time they were developed (with the test collections available the time, which were extremely limited compared to the present generation of TREC-derived material). But the dependence models did not lead to any substantial improvements in performance.

There has been no substantial more recent work on dependence models supported by serious retrieval experiments, and indeed the practical challenges of computing dependencies on the large scale are very considerable. However much of the work done within the TREC Programme on the use of phrases and passages, for instance, can be seen as seeking to capture dependencies by more informal means, though there may be other motivations as well. Thus limiting candidate query expansion terms to those occurring in the passage neighbourhoods of matching terms can be seen as a way of concentrating co-occurrence information so that it is more discriminating than co-occurrences computed over extended full texts would be. Such techniques, as illustrated for example by the Local Context Analysis used with INQUERY (Xu and Croft 1996), have become quite common Sparck Jones (1998b, 1998c), being taken as contributing, if only modestly, to performance.

9.4 Logical information retrieval

More recent work by van Rijsbergen has been in the area of logic and information retrieval, but with a particular probabilistic view incorporated into the logic (van Rijsbergen 1986, Sebastiani 1998). The essence of this approach is to re-interpret the basic concept of relevance as a logical relation between document and query (a document is relevant if it "implies" the query, in a way analogous to theorem-proving). It is then assumed that the information available is in some sense incomplete, so that the implication cannot generally be proved without the addition of missing information. The measure of this missing information is then the probability of relevance.

This work has stimulated a great deal of theoretical discussion, but generally little experimentation, and this only with drastic simplification. The exception is the work of Turtle and Croft and colleagues on the INQUERY system, discussed further below.

9.5 Networks

One class of probabilistic models which has been used extensively in other application areas consists of those based on networks. Such models see the domain of application as a network of nodes, with probabilistic relations between them. Generally the links are taken to represent the important or significant relationships. The absence of a direct link between two nodes is equated either with the absence of any relationship between them, or with the idea that any

such relationship is a secondary one, implied by whatever multistep paths may exist between them.

In the retrieval area, there are several reasons to consider such models. We have plenty of candidate nodes, such as terms, documents and queries, and relationships between them that might be interpreted probabilistically. Indeed, in this paper we have several times appealed to arguments similar to those in the previous paragraph. For example, in the discussion on term frequencies (Section 4), we have supposed the existence of a hidden concept (node) associated with each term (the "eliteness" property), linked to some documents definitely but unknowably, and to the actual occurrences of the terms probabilistically. Furthermore the relation between term occurrences and relevance was assumed to be via the eliteness property, not directly.

Several authors have developed explicit network models for retrieval. The PIRCS system of Kwok is one such (Kwok 1995). This incorporates a "spreading activation" mechanism common to such models, whereby individual nodes are stimulated, and the stimulus spreads through the network via the links. In this case, retrieval involves stimulating a query node and allowing the stimulus to spread via terms to the documents. The most highly stimulated documents are then retrieved.

One component of this mechanism relates closely to the use of blind expansion, discussed in Section 5.3. PIRCS can be set up so that the stimulus passes from query to query terms, to documents indexed by those terms, back to the terms indexing those documents, and back to documents again. This is essentially the same mechanism as blind expansion without relevance information – the documents stimulated first are the initially retrieved items, and the terms stimulated from these documents are the equivalent of our expansion terms.

The INQUERY system of Croft and others (Turtle and Croft (1990, 1991)) is also based on a network approach (as indicated above, their interpretation of probability of relevance is associated with the logical approach to retrieval). In comparison with the model presented in this paper, the inference network model on which INQUERY is based is potentially richer, because additional nodes and links may be incorporated. For example, apart from single query terms, if the query is expressed in Boolean form, the various Boolean constructions can be represented as nodes in their own right. Evidence from different nodes may be combined in different ways: in the Boolean case, this combination can be ade to fit the normal Boolean logic (or extensions of it); or it can follow the sum-of-weights methods used in this paper. Other evidence-combination methods may also be defined.

Both PIRCS and INQUERY have been extensively tested in the TREC Programme; both tend to do well there. Again, there is a multitude of differences between Okapi and either PIRCS or INQUERY, both in matters of principle and in details; but there is also some commonality, and some mutual learning over successive rounds of TREC. As a result it is difficult to attribute the relatively small differences of performance to specific causes.

9.6 Regression

The regression approach was introduced in Section 8.4. Essentially, a regression model takes the form of an assumed relationship between a dependent variable (relevance) and any independent variables which might be significant predictors of the dependent variable. The model then provides methods of estimating the parameters of the model directly from training data.

The probabilistic approach is in some sense complementary to the regression approach, in that regression could be used without reference to the probabilistic model just as the proba-

bilistic model can be used independent of regression (as in this paper). However probabilistic ideas have informed the regression approach in a number of ways. The first is that the dependent variable is generally taken to be *probability* of relevance rather than relevance itself. Second, because of the nature of relevance (assumed binary) and of probability, researchers investigating regression have tended to use logistic regression rather than traditional linear or polynomial regression. Third, some of the forms of relationships assumed in the regression models have been based on those found in probabilistic models (although in this context, regression is eclectic – any form of relationship might be regarded as candidate for a regression approach, whether the form is derived from a model, or observed empirically, or arrived at by any other means).

Despite the previous statement, only a rather limited range of relationships appear to be suitable for regression in practice. For example, it seems that the equations and parameters which are suggested by the probabilistic model (e.g. those involving k_1 or b) are not in a suitable form for learning by regression. On the whole, regression methods have been used to learn about general characteristics of terms, as discussed in Section 8.3 above. Fuhr's group have applied them both to searching and to the indexing stage (see e.g. Fuhr and Buckley (1991), Fuhr et al. (1994)).

Work in this area has had mixed success. One problem is that the optimisation criterion for regression (something like least-squares error) is not necessarily well related to the retrieval performance measures (e.g. Average Precision) by which it is judged. Nevertheless, the Berkeley group (Cooper, Chen and Gey 1994) has had some success at TREC with a version of logistic regression used only when searching.

One observation to emerge from this Berkeley group's work is that if a log-odds score is converted back to an estimate of probability of relevance, documents at the top of the ranking often appear to have absurdly high probabilities. This presumably reflects the inaccuracy of the independence assumptions: combinations of terms do not really imply the kind of overwhelming evidence of relevance that the independence assumptions would suggest. This observation may result in a correction element being applied to the scoring method, which may, depending on its exact form, have little or no effect on the ranking of the documents, but would provide a more accurate estimate of the actual probability of relevance. Such a correction may be of value in the filtering task (see Section 7).

10 Assessment

The object of this section is three-fold. First, we present a summary and overview of the results we have reported, identifying the main conclusions to be drawn about particular strategies and techniques for information retrieval, within the context of the probabilistic model as presented here. Second, we make some general remarks about this probabilistic model and its role in IR theory. Finally, we discuss a range of open issues around the boundaries of the work presented here.

10.1 Test summary and review

We have presented the development of the probabilistic model in successive steps, with accompanying test results. The latter were selected to cover specific points, e.g. whether some particular type of information, used in some particular model-defined way, was advantageous compared with not using this information, in this way. The individual test results are those given in Table 5, and the sequence of comparisons we have made are listed in Table 6 in the Appendix. It is now necessary to consider our set of experiments as a whole, in two ways: first, to take a broader view of the relative, and overall impact on performance of the various data types and retrieval strategies or more specific devices we have detailed; and second to consider the effect of different collection characteristics. We have hitherto been concerned to establish that strategy differences hold across collections and, for our TREC collection, across request forms. However some strategies may be relatively more advantageous under some collection conditions than others, with respect to request or document properties.

As a starting point we give the actual performance figures and significance test data for key runs and comparisons for the TREC collection, covering the major retrieval strategies in exemplar particular instantiations, in Table 3.

Then the overall outcome of our whole series of tests, across data types and strategies, can be summarised as follows. For the older collections, as reported in earlier publications, CFW gives some gain over UW, and RW over CFW. Absolute performance is quite good for RW, even with little relevance information. The same relations hold for TREC, except for the Very short requests, but absolute performance is very low.

TREC is the only collection to which further options apply. So we now focus on it and, considering both Doc30 and Rec30, look for *large* performance differences, which we define here as ones that are at least Noticeable but are typically more than that and hence are likely to be practically useful. This gives the overview shown in Table 4. In the table strategies are grouped, in an informal but intuitive way, into major and subordinate ones, and we also show points about yardsticks.

Thus looking first at the major strategies when we go beyond RW, it is evident that using within-document term frequencies is very valuable, but that using relevance information as well only comes into its own when it is used not for weighting alone, but for expansion: though even here the gain is not very large, or quite complete. In making these summary assessments we are deliberately suppressing performance detail, but that given earlier supports this broad brush picture with the required significance test checking as well as range of more specific comparisons.

With the subordinate strategies on the other hand, there are no large gains to be made by what we may call elaboration, for instance exploiting passage-level matching, giving some extra weight to initial query terms in expansion, or fussing about the precise degree of expansion

or trying to push expansion beyond a moderate level.

The yardstick use of relevance information, on the other hand, confirms the value of this information, subject to the qualification that has to be made about its application with the expansion strategy.

Turning now to absolute performance for the TREC case, and taking $QACIW + E$ (with default expansion) as the best strategy along with the use of rel in 10 as a realistic application of relevance information, we find that Precision at Doc30 is .57 for the best case, with the Long requests, with a corresponding value of .51 for Rec30. Average Precision is .37. This is a very good level of performance.

With respect to the internal differences between request forms for TREC, the main point to note is that expansion is of more value for the Medium requests than the other two: presumably the Long forms do not need it, while it cannot be properly directed from the Very short ones. But more generally, gains from the various strategies explored are least for the Very short requests and are sometimes not Noticeable, though they are for the other forms. This is important because such brief requests are most likely to be encountered in practice. But it is also important that there are gains from the strategies (and hence the model) even for this kind of request.

On the absolute level of performance, taking the same $QACIW + E$ rel in 10 case as previously, we find that the difference between the request forms ranges from .57 for the Long requests, through .52 for Medium to .44 for Very short, with corresponding values for Rec30 of .51, .47 and .37. Average Precision is respectively .37, .34, and .27. Performance for the V requests, though much less impressive than for the Long ones, is still adequate.

Finally, we can assess the merit of the model via the strategies it implies. The largest single all round gains are made by using term frequency information, which is not very original or exciting. But it is also the case that when the best use of relevance information is *directly* compared with not using any, i.e. $QACIW + E$ with $QACW$, there is a further Noticeable gain in performance even for the V requests.

It is normally assumed that using full text compared with only titles+abstracts, or titles alone, on the document side, helps retrieval even with brief requests, but there is no strong evidence for this. So it is unfortunate that the collections we had did not permit systematic comparisons between different request forms for other than full text documents.

10.2 The model's status in IR theory

We have shown how a simple probabilistic model motivates a range of strategies and tactics for the use of certain categories of information within specifc weighting and scoring formulae to be used in retrieval. The experimental results presented have both confirmed the power of these strategies and tactics, and provided suitable values for certain tuning constants that occur in the model.

The probabilistic model is clearly not the only way to approach these issues. Many of our formulae are like (in behaviour if not in form) other formulae, motivated by other theories and/or by pragmatic considerations, that been successfully used in IR. We certainly cannot claim any unique validity or power for the probabilistic model, or for the specific formulae we have presented. However, we believe we have shown that the model provides a good and comprehensible basis for a systematic exposition of the components of the formula and their interrelations.

It is also the case that information retrieval does not depend exclusively on formulae. We

have taken a very simple-minded view of the linguistic, semantic and epistemological issues involved. We rely on the fact, which is very apparent in text retrieval, that the language (English in particular, but not exclusively) allows us to identify content-bearing units relatively easily, and with little concern for the finer issues. Furthermore, these content-bearing units are extremely rich from the point of view of their use in retrieval. This does not, of course, preclude the possibility that more sophisticated approaches in any or all of these areas may provide even richer descriptions.

Within the scope of the probabilistic approach to IR, there are in fact many different (and not always compatible) ideas, concerning both the basic formulation of the model and its development; some of these ideas were discussed in section 9, Comparisons. While the Probability Ranking Principle (section 2 and (Robertson 1977)) is of very general applicability, the model as further developed and presented here cannot be regarded as having an exclusive claim on a justification in probability theory. However, at least part of the usefulness of the present model is that it translates directly into a retrieval mechanism based on a simple query-document matching or scoring function.

All in all, we suggest that the probabilistic model described in this paper is both reasonably well-founded and of clear and substantial value in the design of information retrieval systems.

10.3 Open issues

Query length

While the TREC test data that we have used included queries labelled "very short", we are well aware that this label is somewhat misleading in the context of, for example, searches with Web engines. Here a typical query may be one or two words, one-word queries being very common indeed. It is obvious that many of the techniques we have discussed do nothing at all for one-word queries and very little for two.

One response to this situation is to encourage users to enrich their queries themselves – this approach clearly has very strong justification in terms of the ideas presented here, but its feasibility and likelihood of success are not at all obvious. We must clearly look for other ways of enhancing these queries. One of the strategies discussed here – relevance feedback (actual, not blind) – remains a prime candidate for use in these circumstances, but it still depends on somehow finding one or two relevant documents in the first place.

It also appears to be the case that these queries in practice have very different types of goal and function, both in ther own right and as ways of responding to the mass of heterogeneous material on the Web and some of its distinctive properties, notably URL structure. For instance, an apparently simplistic one-word query may be adequate on a 'just get me something and I'll follow the URLs' basis. Different types of request and need, while familiar to librarians, have not necessarily figured in conventional search systems or in test collection request sets that assume the classical 'need a document about a topic' paradigm. More investigation of request properties is needed, to see whether our model-based approach can respond effectively to requests that are not just short queries, but are also not simple topic ones.

Assigned descriptors

Many databases contain, in addition to free text, index terms of some kind which have been assigned to documents in some more-or-less systematic fashion. These may include for exam-

ple: free text words or phrases, terms taken from a structured thesaurus, codes representing a discrete set of categories, or elements from a fully-structured classification scheme. They may have been assigned by human indexers, or by some automatic process, with or without human intervention.

It seems clear that many of the ideas discussed in this paper apply equally well to such elements as to the types of natural language units discussed. There may however be significant differences (for example the notion of document length may need re-interpreting), and if they are to be combined with text words it may be necessary to distinguish them in some respects. We may also have the situation where an initial query is expressed in terms of such descriptors, or can be mapped diretly onto them; or we may bring them in only at the query expansion stage. In the latter case, we surmise that descriptors may enhance performance by enriching the document descriptions; however, this has not been tested.

Cross-language retrieval

Some similar considerations apply to cross-language retrieval – once we have entered a relevance-feedback mode, the automatic or assisted selection of terms can proceed as with single-language material. However, the initial search is more difficult, unless there are some documents which use both languages, or some translation method for mapping the initial query into the target language. Participants in the TREC cross-language track are investigating such methods, but they have not been tried with the model presented here.

Thresholding and absolute scores

The idea that it might be desirable for a system to produce absolute probability-of-relevance scores, to be compared to an absolute threshold and resulting in a binary retrieval decision, was introduced in Section 7.5, in relation to the filtering task. Actually there may be several situations in which it might be desirable: for example, if the output of the search is to be sent to a further process such as clustering or summarisation, which would require an unranked set of documents. Even if the system is to produce ranked output, it may be regarded as desirable to present the user (or perhaps a subsequent program) with an estimate of the probability of relevance, in which case an absolute score would be required.

For these reasons, we see it as desirable to pursue the question of how to arrive, in such situations, at a reasonable estimate of the absolute probability of relevance. At present the most likely method would seem to be some form of post-hoc calibration of a scoring function, as suggested in Section 7.5.

Other tasks

This paper has been strongly rooted in the conventional view of the retrieval task as a self-contained one, starting with the user presenting with an information need and ending with the user's departure. The discussion immediately above suggests that there may be many other aspects of retrieval in the context of specific wider tasks that would have a bearing on the development, use or interpretation of the model.

Structure and text

Many databases contain elements that might be treated in a structured fashion (in the manner of relational or object-oriented databases). In principle, such elements may be searched using the methods proposed in this paper, although as with assigned indexing, some concepts such as document length may need reinterpretation, and combinations of different term types may need to be tackled. Another constraint is that in some cases, structured elements (e.g. date, or some security classification) may be better seen as providing absolute logical criteria for selection, outside the framework of any probabilistic argument, an idea discussed further below. Certainly the combination of the kinds of retrieval undertaken in structured databases and those appropriate to text deserves and is receiving some attention.

Two-stage retrieval

We can imagine a number of situations in which retrieval might take two stages, either or both of which might involve some of the methods suggested here. For example, we might see retrieval of information from large documents as involving first the retrieval of certain documents in their entirety and then the retrieval of passages from those documents. A second example has just been suggested, where an initial logical criterion is applied and a subsequent probabilistic search is conducted on the resulting set.

Such ideas introduces a number of considerations into the interpretation of the elements of the probabilistic model. For example, we have generally assumed that the document collection was likely to be large, orders of magnitude larger than the number of documents that a user would be prepared to look at. However, if an initial logical criterion is applied, it may well be that the resulting set is not so large. This fact may for example invalidate the practice indicated in Section 4.2, of regarding all documents not yet known to be relevant as non-relevant.

The possibly small size of the set within which further analysis is to be applied has other consequences. In section 4 it was asserted that the traditional CFW and the Croft/Harper (1979) approximation based on RW were very similar. This is true in the normal situation where any particular term is expected to occur in only a small fraction of a large collection, but not otherwise. In fact if a term occurs in over half the set under consideration, the Croft/Harper weight is negative. This suggests a closer look at the behaviour of the model under such circumstances. We might also be interested in identifying terms which *describe* individual documents in a selected set of documents (as in the mini-summarising task discussed in Section 7), or which describe sets of documents for a user: the description requirement which deals with document(s) in their own right has some similarities to, but also some differences from, the selection one which motivates attempts to find terms that distinguish specific documents from others in a collection; and these differences would appear to be magnified in situations with a small base set.

Other media

We may consider for retrieval documents in media other than text, for example images, spoken material, other sounds, and again the question arises: would the models and methods discussed here be appropriate for such material? Before attempting an answer, some observations are appropriate.

One of the characteristics of textual material, which is of major significance in this context, is that it is relatively easy to discover meaningful content-bearing units as the raw material for indexing. Specifically, whatever the problems associated with them, words are actually extremely valuable elements for this purpose. Even if we consider languages other than English, in which word segmentation may not be so straightforward, the general comment still applies. Images or sounds, by contrast, provide no such obvious meaning-bearing primitive elements.

While we could consider doing relevance feedback using any kind of identifiable elements at all, it is also the case that it will not work very well unless the set of elements available include at least some which are likely to be correlated with relevance. Thus while one could index an image by its individual pixels, and possibly discover some useful correlations with relevance for some queries, in general we would not expect to do very well that way. A further problem is that of the initial query, as with cross-language retrieval.

It seems likely that any useful application of the methods presented here to such material will depend on the identification of higher-level features which might reasonably be assumed to be content-bearing. We would be dependent on the use of other systems to extract such features. That said, if we have features which can be used in this way (possibly a variety of features, of different types and/or from different sources), the methods we have advanced should work with them.

Table 3: Key runs and comparisons, TREC T741000X collection

(figures rounded)

	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
RW pred top 3	.16	.21	.18	.13	.18	.18	.08	.12	.12
QACW	.50	.44	.40	.44	.37	.34	.32	.27	.24
QACIW pred rel in 10	.52	.46	.42	.46	.42	.36	.33	.28	.25
QACIW + E pred rel in 10	.57	.52	.43	.51	.47	.36	.37	.34	.26
(exp L=32, M =24 V=16)									

Significance tests, Wilcoxon

s = significant at the 1% level

m = significant at the 2.5% level

X = not significant

CFW	vs	UW	s	s	s	s	s	s	s	s	s
RW		CFW	s	s	s	s	s	s	s	m	s
CW		CFW	s	s	s	s	s	s	s	s	s
QACW		CFW *									
QACIW		QACW	s	s	s	s	m	s	s	s	s
QACIW + E		QACIW	s	s	s	s	s	s	x	x	x

*not run, use CW vs CFW as given

Table 4: Overview of results, TREC T741000X collection

all request forms
large performance differences at least Noticeable, typically more

	exceptions
1) major strategies	
CFW large gain on UW	
RW large gain on CFW	V
(QA) CW very large gain on CFW	
QACIW very large gain on RW	
QACIW no gain on QACW if little rel info	
QACIW + E large gain on QACIW even if little rel info	V
QACIW + E no gain on QACIW with blind 'rel'	
2) subordinate strategies	
QACIW + E heavy expansion no gain	
QACIW + E query term emphasis no gain	
QACIW, QACIW + E passage matching no gain	
3) yardsticks	
RW retro large gain on pred if pred little rel	
QACIW retro large gain on pred if pred little rel	
QACIW + E retro large gain on pred if pred little rel	L

References

Biebricher, B., Fuhr, N., Lustig, G. Schwanter, M. and Knorz, G. (1988) The automatic indexing system AIR/PHYS - from research to application. *Proceedings of the 11th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval* New York: Association for Computing Machinery, 333-342.

Buckley, C., Allan, J. and Salton, G. (1995) Automatic retrieval and routing using SMART: TREC-2. *Information Processing and Management*, 31, 315-326.

Buckley, C., Singhal, A. and Mitra, M. (1996) New retrieval approaches using SMART: TREC-4. *The Fourth Text REtrieval Conference (TREC-4)*, (Ed. D.K. Harman) Special Publication 500-236, National Institute of Standards and Technology, Gaithersburg, MD, 25-48.

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

Cooper, W., Chen, A. and Gey, F. (1994) Full text retrieval based on probabilistic equaitions with coefficients fitted by logistic regression. *The Second Text REtrieval Conference (TREC-4)*, (Ed. D.K. Harman), Special Publication 500-215, National Institute of Standards and Technology, Gaithersburg, MD, 57-66.

Cooper, W.S. and Maron, M.E. (1978) Foundations of probabilistic and utility-theoretic indexing. Journal of the ACM, 25, 67-80.

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

Callan, J.P., Croft, W.B. and Broglio, J. (1995) TREC and TIPSTER experiments with INQUERY. *Information Processing and Management*, 31, 327-343.

Evans, D.A, and Lefferts, R.G. (1995) CLARIT-TREC experiments. *Information Processing and Management*, 31, 385-395.

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

Fuhr, N., Pfeifer, U. Bremkamp, C, and Pollman, M. (1994) Probabilistic learning approaches for indexing and retrieval with the TREC-2 collection. *The Second Text REtrieval Conference (TREC-4)*, (Ed. D.K. Harman), Special Publication 500-215, National Institute of Standards and Technology, Gaithersburg, MD, 1997, 67-74.

Hand, T.F. (1997) A proposal for task-based evaluation of text summarisation systems. In *Intelligent, scaleable text summarisation* (Ed. I. Mani and M. Maybury), Proceedings of a Workshop, Somerset, NJ: Association for Computational Linguistics, 1997, 31-38.

Harman, D.K., also Voorhees, E.M. and Harman, D.K. (1993-1997) Overviews of TREC Conferences in successive Proceedings, See TREC (1993-1997), National Institute of Standards and Technology, Gaithersburg, MD.

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.

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

Harter, S.P. (1975) A probabilistic approach to automatic keyword indexing. Parts 1 and 2. *Journal of the American Society for Information Science*, 26, 197-206 and 280-289.

Hawking, D. and Thistlethwaite, P. (1998) Overview of TREC-6 very large collection track. *The Sixth Text REtrieval Conference (TREC-6)*, Special Publication 500-ZZZ, National Institute of Standards and Technology, Gaithersburg, MD, in press.

Ide, E. (1968) New experiments in relevance feedback. In *Scientific Report ISR-14*, Cornell University. Reprinted as Chapter 16 in *The SMART retrieval system* (Ed. Salton), Englewood Cliffs, NJ: Prentice-Hall, 1971.

Jones, G.J.F., Foote, J.T., Sparck Jones, K. and Young, S.J. (1996) Retrieving spoken documents by combining multiple evidence sources. *SIGIR 96: Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, New York: Association for Computing Machinery, 30-38.

Kwok, K.L. (1995) A network approach to probabilistic information retrieval. *ACM Transactions on Information Systems*, 13, 325-353.

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

Lewis, D. (1997) The TREC-5 filtering track. In *The Fifth Text REtrieval Conference (TREC-5)* (Ed. E.M. Voorhees and D.K. Harman), Special Publication 500-238, National Institute of Standards and Technology, Gaithersburg, MD, 75-96.

Mani, I. and Maybury, M. (Eds.) (1997) *Intelligent, scaleable text summarisation*, Proceedings of a Workshop, Somerset, NJ: Association for Computational Linguistics, 1997.

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

Mitra, M., Buckley, C., Singhal, A. and Cardie, C. (1997) An analysis of statistical and syntactic phrases. *Proceedings of RIAO-97, Computer-Assisted Information Searching on Internet*, Centre de Hautes Etudes Internationales d'Informatique Documentaires, Paris.

Okapi (1997) Papers on Okapi, Special Issue of *Journal of Documentation*, 33, 3-87.

Peat, H.J. and Willett, P. (1991) The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science*, 42, 378-383,

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

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.

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

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

van Rijsbergen, C.J. (1986) A non-classical logic for information retrieval. *The Computer Journal*, 29, 481-485.

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

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

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

Robertson, S.E. and Bovey, J.D. (1982) Statistical problems in the application of probabilistic models to information retrieval. Technical Report, Centre for Information Science, City University.

Robertson, S.E. and Harding, P. (1984) Probabilistic indexing by learning from human indexers. *Journal of Documentation*, 40, 264-270.

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., van Rijsbergen, C.J. and Porter, M.F. (1981) Probabilistic models of indexing and searching. In *Information retrieval research* (Ed. W.R. Oddy et al.). London: Butterworths, 35-65.

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. (1994) Some simple effective approximations to the 2 Poisson model for probabilistic weighted retrieval. *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York: Association for Computing Machinery, 232-241.

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.

Rocchio, J.J. (1965) Relevance feedback in information retrieval. In *Scientific Report ISR-9*, Harvard University. Reprinted as Chapter 14 in *The SMART retrieval system* (Ed. G. Salton). Englewood Cliffs, NJ: Prentice-Hall, 1971.

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.

Salton, G., Singhal, A. Mitra, M. and Buckley, C. (1997) Automatic text structuring and summarisation. *Information Processing and Management*, 33, 193-207.

Sebastiani, F. (1998) On the role of logic in information retrieval. *Information Processing and Management*, 38 (1), 1-18.

Pivoted document length normalisation. *Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval* New York, Association for Computing Machinery, 11-29.

Smeaton, A. and Wilkinson, R. (1997) Spanish and Chinese document retrieval in TREC-5. In *Proceedings of the Fifth Text REtrieval Conference (TREC-5)* (Ed. E.M. Voorhees and D.K. Harman), Special Publication 500-238, National Institute of Standards and Technology, Gaithersburg, MD, 57-64.

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. (1980) Search term relevance weighting - some recent results, *Journal of Information Science*, 1, 325-332.

Sparck Jones, K. (1998) (1998a) What is the role of NLP in text retrieval? In *Natural language information retrieval* (Ed. T. Strzalkowski), Dordrecht: Kluwer, in press.

Sparck Jones, K. (1998) (1998b) Summary performance comparisons: TREC-2, TREC-3, TREC-4, TREC-5, TREC-6. In *The Sixth Text REtrieval Conference (TREC-6)*, Special Publication 500-ZZZ, National Institute of Standards and Technology, Gaithersburg, MD, in press.

Sparck Jones, K. (1998) (1998c) Further reflections on TREC. Prepared for *Information Processing and Management*, in press.

Sparck Jones, K., Jones, G.J.F., Foote, J.T. and Young. S.J. (1996) Experiments in spoken document retrieval. *Information Processing and Management*, 32, 399-419.

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

Sparck Jones, K. and Willett, P. (Eds.) (1997) *Readings in information retrieval*, San Francisco: Morgan Kaufmann.

TREC (1992-1997): 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, NIST, 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)* (1997), in press.

Turtle, H.R. and Croft, W.B. (1990) Inference networks for document retrieval. *Proceedings of the 13th International ACM-SIGIR Conference on Research and Development in Information Retrieval*, New York: Association for Computing Machinery, 1-24, 1990.

Turtle, H.R. and Croft W.B. (1991) Evaluation of an inference network-based retrieval model, *ACM Transactions on Information Systems*, 7, 187-222.

Walker, S., Robertson, S.E., Boughanem, M., Jones, G.J.F., and Sparck Jones, K. Okapi at TREC-6: automatic ad hoc, VLC, routing, filtering and QSDR. *Proceedings of the Sixth Text REtrieval Conference (TREC-6)*, Special Publication 500-ZZZ, National Institute of Standards and Technology, Gaithersburg, MD, in press.

Xu, J. and Croft, W.B. (1996) *Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval* New York, Association for Computing Machinery, 4-11.

Appendix

This appendix gives detailed retrieval results and significance test data.

Table 5 shows first, the run output for the TREC T741000X collection, in fact for the odd-numbered Half collection. This allows for predictive relevance weights that have been computed using the even-numbered other half; retrospective relevance weights are computed on the odd-numbered documents.

The TREC collection 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. NOTE that the values for Document Cutoff at 30, labelled P30, are those named Doc30 in the body of the paper, and that Precision at Recall 30, P30R, is called Rec30 in the body of the paper. The figures are simply truncated.

The latter part of the table 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 6 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 5: Retrieval run results: T741000X collection, L, M and V requests; C1400I, U27000P, N11500A collections

Long requests

	AveP	P5	P10	P15	P20	P30	P100	RPrec	P30R	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
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 5 (contd): Retrieval run results

Medium requests

	AveP	P5	P10	P15	P20	P30	P100	RPrec	P30R	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
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 5 (contd): Retrieval run results

Very short requests	AveP	P5	P10	P15	P20	P30	P100	RPrec	P30R	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
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 5 (contd): Retrieval run results

Old collections results : all collections Half ; e = guestimated

	AveP	P5	P10	P20	P30	P100	P30R	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 6: Significance test results, selected runs, T741000X collection

```
Wilcoxon signed ranks test: - means < 1.96, . means 1.96 - 2.33, + means > 2.33
 1.96 and 2.33 correspond to 1% and 2.5% 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 retro                  RW pred all     >       + + +   + + -   + . -
RW retro                  RW pred top 3   >       + + +   + + +   + + +
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 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 e16   QACIW+E pred r/10 e24 =                 - - .
S'QACIW+E pred r/10 e16   QACIW+E pred r/10 e32 =         - - -
L QACIW+E pred r/10 e16   QACIW+E pred r/10 e48 >       + + +
L QACIW+E pred r/10 e16   QACIW+E pred r/10 e72 =       + + +

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     =       + - -   + - -   + - -
```