Exploring the Similarity Space

Justin Zobel* Alistair Moffat†

Abstract *Ranked queries are used to locate relevant documents in text databases. In a ranked query a list of terms is specified, then the documents that most closely match the query are returned—in decreasing order of similarity—as answers. Crucial to the efficacy of ranked querying is the use of a similarity heuristic, a mechanism that assigns a numeric score indicating how closely a document and the query match. In this note we explore and categorise a range of similarity heuristics described in the literature. We have implemented all of these measures in a structured way, and have carried out retrieval experiments with a substantial subset of these measures.*

Our purpose with this work is threefold: first, in enumerating the various measures in an orthogonal framework we make it straightforward for other researchers to describe and discuss similarity measures; second, by experimenting with a wide range of the measures, we hope to observe which features yield good retrieval behaviour in a variety of retrieval environments; and third, by describing our results so far, to gather feedback on the issues we have uncovered. We demonstrate that it is surprisingly difficult to identify which techniques work best, and comment on the experimental methodology required to support any claims as to the superiority of one method over another.

1 Introduction

It is now commonplace for large document databases to be queried using content-based *ranked* queries, and generally accepted that the alternative mechanisms for searching for information (such as Boolean queries and hierarchical subject descriptors) do not in general provide the same levels of retrieval effectiveness. The implementation of ranked querying is also now well understood [Frakes and Baeza-Yates 1992; Witten et al. 1994; Korfhage 1997], and as a consequence of the spur provided by the TREC project [Harman 1995] there are several publicly-available retrieval systems that support fast ranking on document collections in the multi-gigabyte range.

This leads to the question as to what similarity calculation should be used for each type of query, or type of document, or type of desired performance (high precision versus high recall being one obvious distinction); or even whether such categorisations are possible or meaningful. It is extremely difficult—as illustrated by the results of this paper—to identify a single all-encompassing "best" similarity measure, and we do not propose one here. What we do observe, however, is that there has been convergence towards a small number of good measures, in particular, those that perform well in the TREC environment, and that there is considerable doubt as to what components of those formulae are responsible for the good performance.

In this note we take a fresh look at the various facets of a similarity measure, proposing an eight-way orthogonal decomposition into factors that in one form or another appear consistently in most of the measures we have found in the literature. The decomposition allows similarity measures to be specified as points in the eight-space, and so permits the space of similarity measures to be explored in systematic and coherent manner. We believe that our eight-way categorisation is sufficiently general that most measures can be described in the same framework.

Having identified these eight components we are able to regard each component as a dimension that can be explored. To allow each formulation to be tested we extended the public-domain text database system mg [Moffat and Zobel 1994; Witten et al. 1994; Bell et al. 1995; MG-software 1995] to permit each of the eight

*Department of Computer Science, RMIT, GPO Box 2476V, Melbourne 3001, Victoria, Australia. Email: jz@cs.rmit.edu.au

†Department of Computer Science, the University of Melbourne, Parkville, 3052, Melbourne, Victoria, Australia. Email: alistair@cs.mu.oz.au

components to be modified independently. That is, we developed a version of mg that, at database creation time, is parameterised by a *Q-expression*, an eight-position string specifying the similarity computation to be performed. At query time the mg system now allows a Q-expression to be specified; if the various index and weights files required to evaluate the specified Q-expression have not been created then the software will print as an error diagnostic the command line or lines that should be executed to create the necessary files.

Using this version of mg we explored a large number of variant measures to test whether particular formulations for some components work well regardless of the combination in which they are used, and whether there are new combinations that are more effective than the measures in common use. We used two test collections and three sets of queries, giving six experimental domains. The breadth of experimentation was deliberate; one of the goals of the investigation was to measure the extent to which good performance in one domain implies good performance in another.

Intrinsic to these experiments is the notion that performance can be compared in a reliable way. A standard method of comparison is to use a recall-precision average, but in the context of the TREC data recall-precision cannot be completely evaluated because the number of relevant documents is unknown [Zobel 1998]. Recall-precision is even less reliable when used to gauge subcollections as there are proportionately fewer relevance judgements. For this reason we also used as measures of performance the precision at a fixed (and relatively small) number of documents retrieved, and the rank of the first relevant document retrieved. These latter measures are important when, for example, a single screenful of top-ranked documents is to be returned to a user as the answer to an information search.

We expected in this experimental phase of our investigation to confirm that standard formulations of similarity measures are effective, and indeed this is what occurred. What was surprising, however, was that there was no overall winner, and most of the techniques that worked well in one of the six experimental domains worked poorly in at least one of the other five. Nor did good performance according to one metric necessarily correspond to good performance according to another. The results of the experiments were sufficiently contradictory that we used statistical tools to ensure that the variations being observed were the reflection of genuine differences in behaviour, and not the result of random fluctuation.

2 Similarity Measures

Many similarity measures have been proposed, based on the vector space model and the probabilistic model as well as naive co-occurrence of terms between document and query. In this section we describe several standard similarity measures, mostly based on the vector space model, in a consistent framework and notation. We have used a range of sources but are particularly indebted to van Rijsbergen [1979]; Salton and McGill [1983]; Salton [1989]; and the various authors that contributed to Frakes and Baeza-Yates [1992]. Interested readers are referred to the bibliographies of these books for citations to original publications. We have also made use of the proceedings of the first five TREC conferences.

Our work is, in many ways, an extension of a previous taxonomy due to Salton and Buckley [1988]. They examined a variety of ways for assigning weights to terms in documents and queries, supposing throughout that the cosine combining mechanism was used to derive a final similarity score. Salton and Buckley used five smaller test collections—the best that were available prior to the TREC initiative—and listed results for eight good combinations. In their six-dimensional space they describe alternative formulations that admit $(3 \times 3 \times 3)^2 = 729$ different similarity measures. The placement of their methods in our scheme is discussed further below.

Atomic components A *collection* is a body of information, usually, but by no means always, consisting of text. A *document* is the smallest unit of access within the collection, for example, one newspaper article. A *term* is some identified concept within a document. For text documents the terms are commonly taken to be the words of the document, after stemming and similar transformations; but a term might also be a word pair, a phrase, or an externally assigned descriptor that does not appear in the document at all.

A general property of almost all similarity measures is that each is a combination of simple statistics, or primitive information, about the document collection, including:

- the number N of documents;

- the number n of distinct terms used in the collection;
- for each term t and each document d containing t, the frequency $f_{d,t}$ of t in d;
- for each term t, the total number F_t of occurrences of t in the collection;
- the number f_t of documents containing term t;
- for each term t, the frequency $f_{q,t}$ of t in query q;
- for each document d, the value $f_d = |d|$, the number of term occurrences in d;
- for each document d, f_d^m, the largest $f_{d,t}$ of any term in d; and
- f^m, the largest f_t in the collection.

Documents and terms are then gathered into sets that restrict the domain of the operations used to combine the statistics into similarity values. We denote these various sets as:

- the set $\mathcal{D}$ of documents;
- for each term t, the set $\mathcal{D}_t$ of documents containing t;
- the set $\mathcal{T}$ of distinct terms in the database; and
- the set $\mathcal{T}_d$ of distinct terms in document d, and similarly $\mathcal{T}_q$ for queries, and $\mathcal{T}_{q,d} = \mathcal{T}_q \cap \mathcal{T}_d$.

Thus $f_t = |\mathcal{D}_t|$ and $F_t = \sum_{d \in \mathcal{D}_t} f_{d,t}$. Note that $f_d \geq |\mathcal{T}_d|$.

The basic statistics are combined in different ways by different similarity measures, and are detailed below. There are, however, three important monotonicity assumptions that are present in all formulations, and it is worth stating these explicitly. They are that rare terms are no less important than frequent terms; that multiple appearances of a term in a document are no less important than single appearances; and that, for the same quantity of term matching, long documents are no more important than short documents.

Combining functions The *similarity* $S_{q,d}$ of a document to a query, which we refer to as the *combining function*, is usually derived from $w_{d,t}$, $w_{q,t}$, W_d, and W_q, which correspond respectively to the importance of each term in the document, the importance of that term in the query, the length or weight of the document, and the length of the query. All of these quantities are defined in detail below. The similarity measures $S_{q,d}$ we consider are shown in Table 1. In all cases $S_{q,d}$ is intended to numerically indicate how close the document d and query q are in their information content. High scores indicate substantial overlap in term usage, and low scores indicate dissimilar term usage. When $\mathcal{T}_{q,d}$ is empty all of these $S_{q,d}$ formulations yield zero. The formulations have been assigned alphabetic labels for later reference.

Term weight Terms that appear in many documents in the collection should be discounted compared with terms that appear in only a few documents. Thus, it is usual to take into account a *term weight* (also known as an *inverse document frequency* or *IDF*), denoted here as w_t. Many methods have been suggested for calculating term weight; the formulations we consider are shown in Table 2. Term discrimination [Salton and McGill 1983] is also of interest, but we are not aware of a practical method of computing it. Salton and Buckley [1988] describe three different term weighting rules, and also allow the weight to differ between the document and the query. Their three weighting rules are noted in Table 2 with the acronym "SB" and the code assigned by Salton and Buckley to that combination.

Document-term and query-term weights Given a term weight, the next decision is the specification of where it should be used—in constructing the *document-term weight* denoted $w_{d,t}$, the *query-term weight* denoted $w_{q,t}$, in neither, or in both. When it is used it biases the relative term frequency $r_{d,t}$, defined in the next paragraph. The two alternative methods for doing this are shown in Table 3.

The quantities $w_{d,t}$ and $w_{q,t}$ are derived from other calculated values; nevertheless, it is useful to distinguish them. Doing so allows whichever formulation of w_t is chosen to be applied selectively to either or both of query terms and document terms. This allows a wider range of possibilities than when $w_t = 1$ is used, the first formulation in Table 2.

	Description	Formulation
A	Inner product.	$S_{q,d} = \sum_{t \in \mathcal{T}_{q,d}} (w_{q,t} \cdot w_{d,t})$
B	Cosine measure.	$S_{q,d} = \frac{\sum_{t \in \mathcal{T}_{q,d}} (w_{q,t} \cdot w_{d,t})}{W_q \cdot W_d}$
C	Simple probabilistic measure. The variable C is a tuning constant, set to 0 in this context [Frakes and Baeza-Yates 1992, p. 369].	$S_{q,d} = \sum_{t \in \mathcal{T}_{q,d}} (C + w_t)$
D	More sophisticated probabilistic measure. Variable C is again a tuning constant set to 0.	$S_{q,d} = \sum_{t \in \mathcal{T}_{q,d}} (C + w_t) \cdot r_{d,t}$
E	Alternative inner product.	$S_{q,d} = \sum_{t \in \mathcal{T}_{q,d}} \frac{w_{d,t}}{W_d}$
F	Dice formulation. (Ozkarahan [1986, p. 496] and Salton and McGill [1983, pp. 202–3] use $W_x = \sum_{t \in \mathcal{T}_x} w_{x,t}$ rather than W_x^2, for Dice, Jaccard, and overlap.)	$S_{q,d} = \frac{2 \sum_{t \in \mathcal{T}_{q,d}} (w_{q,t} \cdot w_{d,t})}{W_q^2 + W_d^2}$
G	Jaccard formulation.	$S_{q,d} = \frac{\sum_{t \in \mathcal{T}_{q,d}} (w_{q,t} \cdot w_{d,t})}{W_q^2 + W_d^2 - \sum_{t \in \mathcal{T}_{q,d}} (w_{q,t} \cdot w_{d,t})}$
H	Overlap formulation.	$S_{q,d} = \frac{\sum_{t \in \mathcal{T}_{q,d}} (w_{q,t} \cdot w_{d,t})}{\min(W_q^2, W_d^2)}$

Table 1: Combining functions $S_{q,d}$

Relative term frequency Most ranking rules attempt to emphasise the effect of terms that are frequent in either document or query or both. This quantity—denoted $r_{d,t}$ in the document and $r_{q,t}$ in the query—is known as the *relative frequency*, or relative term frequency. The formulations of relative frequency we consider are shown in Table 4. The table shows $r_{d,t}$; values of $r_{q,t}$ are calculated in a corresponding manner based upon $f_{q,t}$. Either (or both) of these values are sometimes known as the *TF* component, and so similarity formulations that are described as TF-IDF make use of a relative term frequency and an inverse document frequency somewhere in their calculation. Salton and Buckley [1988] made use of three different relative term frequency formulations; these are noted in the table.

Document and query length Some formulations of document length W_d and query length W_q are shown in Table 5, which are often (but not always) derived from the $w_{d,t}$ and $w_{q,t}$ values respectively. Here the desire is to allow for long documents, which may contain many appearances of query terms but be no more relevant than a succinct document containing only a few appearances. Thus, W_d and W_q are used in the calculation of the combining function (Table 1); Table 5 describes some of the many possible ways of quantifying the length of a document. For example, the first formulation might be of use when documents are known to be of fairly uniform length, perhaps in a bibliographic retrieval system that stores title and author but not abstract. The last formulation—the pivoted method—is a means of adjusting the cosine method to account for experimentally-determined bias [Singhal et al. 1996], and can be applied to any of the non-unit measures. For orthogonality we also list pivoting as being applicable to the unit length calculation. Analogous formulations are used for W_q, but note that there is no query-length equivalent of pivoting. Salton and Buckley [1988] described two length calculations for each of documents and queries.

	Description	Formulation
A	Formulation used for binary match. $SB = x$	$w_t = 1$
B	Logarithmic formulation. $SB = f$	$w_t = \log_e\left(1 + \frac{N}{f_t}\right)$
C	Hyperbolic formulation.	$w_t = \frac{1}{f_t}$
D	Normalised formulation.	$w_t = \log_e\left(1 + \frac{f^m}{f_t}\right)$
E	Another normalised formulation. $SB = p$	$w_t = \log_e \frac{N - f_t}{f_t}$
	These formulas define noise and entropy, where n_t is the noise of t and s_t is the signal.	$n_t = \sum_{d \in \mathcal{D}_t}\left(-\frac{f_{d,t}}{F_t}\log_2\frac{f_{d,t}}{F_t}\right)$ $s_t = \log_2(F_t - n_t)$
	Using these measures, possible definitions of w_t are as shown. The last of these is the entropy measure.	F. $w_t = s_t$ G. $w_t = \frac{s_t}{n_t}$ H. $w_t = \left(\max_{t' \in T} n_{t'}\right) - n_t$ I. $w_t = 1 - \frac{n_t}{\log_2 N}$

Table 2: Term weights w_t (inverse document frequencies)

	Description	Formulation
A	TF-only formulation.	$w_{d,t} = r_{d,t}$
B	Standard formulation, TF-IDF.	$w_{d,t} = r_{d,t} \cdot w_t$

Table 3: Document-term weights $w_{d,t}$ and query-term weights $w_{q,t}$

	Description	Formulation
A	Formulation used for binary match. $SB = b$	$r_{d,t} = \begin{cases} 1 & \text{if } t \in T_d \\ 0 & \text{otherwise} \end{cases}$
B	Standard formulation. $SB = t$	$r_{d,t} = f_{d,t}$
C	Logarithmic formulation.	$r_{d,t} = 1 + \log_e f_{d,t}$
D	Normalised formulation.	$r_{d,t} = \frac{f_{d,t}}{f_d^m}$
E	Alternative normalised formulation. Variable K is a tuning constant, with reported optimums 0.3 and 0.5 [Frakes and Baeza-Yates 1992, page 370]. A similar formulation can be used for query terms, with f_q^m in the denominator [Frakes and Baeza-Yates 1992, page 375]. In our experiments $K = 0.5$ was used. $SB = n$	$r_{d,t} = K + (1 - K)\frac{f_{d,t}}{f_d^m}$
F	Okapi formulation [Robertson et al. 1995]. Not defined for query terms.	$r_{d,t} = \frac{f_{d,t}}{f_{d,t} + W_d / av_{d \in D}(W_d)}$

Table 4: Relative term frequencies $r_{d,t}$ and $r_{q,t}$

Putting it all together Tables 1 to 5 provide for a bewildering array of query-document similarity measures. There are eight possible combining functions (Table 1), nine ways of choosing term weights (Table 2), two ways of choosing document-term weights and query-term weights (Table 3), six ways of setting relative term frequencies for the document and another five for the relative term frequencies in the query (Table 4), and so on. These options are summarised in Table 6, which describes the composition of an eight-character *Q-expression* that selects one possible combination of options. Table 6 also gives as an example, the calculation that corresponds to the Q-expression `BB-ACB-BAA`, where the hyphens in the Q-expression serve to separate the global selection of combining method and term weight from the three factors that affect each of document terms (in the middle group) and query terms (in the final group).

The simplest Q-expression is `AA-AAA-AA`, which scores a document according to the number of the query terms that are present without incorporating any frequency or weight information—an approach sometimes known as co-ordinate matching. In a similar way, the Q-expression `BB-BBB-BBB` describes a method using the cosine similarity coefficient, TF-IDF calculation of term weights in both document and query, and document and query weights calculated as the Euclidean length of vectors in n-space—in other words, the traditional stock-standard cosine vector-space rule. The best formulation investigated by Salton and Buckley—in their terms the measure $tfc \cdot nfx$—would be described in our notation by the Q-expression `BB-BBB-BEA`.

Each other 8-character Q-expression describes a variation. In total there are more than 1,500,000 combinations. However, not all choices give distinct measures. For example, all methods `AA-A**-A**` (where a * indicates an unspecified or "doesn't matter" position) are mathematically identical, since combining function A is independent of both W_d and W_q, and $w_t = 1$ makes the two alternative formulations for both $w_{d,t}$ and $w_{q,t}$ identical. Similarly, some of the formulations are logically identical, since they result in the same ranking. For example, in the cosine formulations `BB-BBB-BB*` the query weight (which for a given query is a constant) serves only to scale the final similarity values, and so does not alter the effectiveness score for a particular experiment. There are other families of equivalent formulations, and, while we have not attempted to exactly determine the number of different measures, estimate it to be of the order of 100,000.

	Description	Formulation
A	Unit length. $SB = x$	$W_d = 1$
B	Vector space formulation. $SB = c$	$W_d = \sqrt{\sum_{t \in T_d} w_{d,t}^2}$
C	Approximate formulation.	$W_d = \lvert T_d \rvert$
D	Another approximate formulation.	$W_d = \sqrt{\lvert T_d \rvert}$
E	Yet another approximate formulation.	$W_d = \log_2 \lvert T_d \rvert$
F	Byte size. (Alternatively, $W_d = b_d$ can be used, where b_d is the length of d in bytes.)	$W_d = f_d$
G	Further alternative approximate formulation.	$W_d = \sqrt{f_d}$
H–N	Pivoted cosine method (used only for document weights), where W'_d is calculated using another length formulation such as method A (to get method H) or method B (to get method I) and where s is the slope, typically about 0.7.	$W_d = (1 - s) + s \cdot \frac{W'_d}{av_{d \in D} W'_d}$

Table 5: Document lengths W_d and query lengths W_q

3 Experiments

We now describe the experimental investigation that was pursued. Our aim was to identify successful combinations—similarity measures that give good effectiveness—and to determine whether there were components, such as particular term weightings, that worked well in all combinations. The space of similarity measures can be partially explored by the various Monte Carlo search mechanisms such as simulated annealing, genetic algorithms, and so on. These methods rely upon the presence of continuity in the objective function, so that "better" solutions are likely to be found in the neighbourhood of "already good" solutions. In early experiments we used a primitive hill-climbing method to attempt to search for the best combinations, but it became clear that the space of similarity measures had a highly irregular topology, and the assumption that immediate neighbours of a combination would have similar performance is unwarranted. Changing any element of a combination can have, and indeed usually does have, a non-trivial effect.

Another method of search is to undertake an exhaustive enumeration, examining every combination. For small search spaces this is tractable, but for large spaces it is implausibly expensive. Each of the experiments we sought to run would require 5–10 minutes of CPU time for each query set to be tested, and an exhaustive search over even 100,000 similarity heuristics with six experiments to run would thus take as long as ten years. We chose instead to investigate a subspace of combinations that were likely to work well, limiting the dimensions of the search space to make exhaustive evaluation possible. These limits were chosen by identifying a small number of measures that are known to be effective, then allowing all possible mix-and-match combinations that could be derived from these measures. In imposing such limits we have, of course, closed off parts of the search space. Some of the excluded regions will be explored in future work.

Component	Position	Number of Variants	Sample Q-expression	Sample Calculation
Combining function for document d and query q:				
$S_{q,d}$	1	8	B	$S_{q,d} = \left(\sum_{t \in d \cap q} w_{q,t} \cdot w_{d,t}\right) / (W_d \cdot W_q)$
Weight of term t:				
w_t	2	9	B	$w_t = \log_e(1 + N/f_t)$
Weight of term t in document d:				
$w_{d,t}$	3	2	A	$w_{d,t} = r_{d,t}$
Relative frequency of term t in document d:				
$r_{d,t}$	4	6	C	$r_{d,t} = 1 + \log_e f_{d,t}$
Weight of document d:				
W_d	5	7 + 6	B	$W_d = \sqrt{\sum_{t \in d} w_{d,t}^2}$
Weight of term t in query q:				
$w_{q,t}$	6	2	B	$w_{q,t} = r_{q,t} \cdot w_t$
Relative frequency of term t in query q:				
$r_{q,t}$	7	5	A	$r_{q,t} = 1$
Weight of query q:				
W_q	8	7	A	$W_q = 1$

Table 6: Example similarity measure BB-ACB-BAA

Choice of similarity formulations We fixed one of the eight dimensions to what we believed to be a reasonable value, and tested a subset of each of the other seven dimensions. The search space explored is described by the regular expression

`[AB] [BDI] - [AB] [CEF] [BDIK] - [AB] [ACE] A .`

That is, we considered combining functions A and B (Table 1); term weight formulae B, D, and I (Table 2); and so on. The only factor that was fixed was the query length W_q. This generated a total of 720 legal mechanisms,[1] containing several mechanisms that are known to work well and a good proportion of mechanisms that are at least in theory likely candidates. At about 5–10 minutes per mechanism for each of the six experimental domains (2–4 seconds per query per collection) our experiments took about four weeks of computation.

We made the search-space restrictions with some trepidation, but also in the belief that doing so would not substantially handicap our findings. For example, we expected the cosine combining function (a B in the first position) to be important, in which case the mechanism used for calculating the query weight has no effect upon the performance. Most of the mechanisms that have been successful at TREC do lie within the space we explored. We also included pivoting [Singhal et al. 1996] and several variant mechanisms for calculating $S_{q,d}$ and w_t.

Database and relevance judgements We used two databases in our experiments. In order to have the maximum number of queries that could be tested, we focussed on the collections on disk 2 of the TREC data. Disk 2 has been part of the testing for several years, and there are 300 queries for which disk 2

[1] Combinations `**-*FB-***` and `**-*FI-***` are not viable, since the average document length is used. This is why the number of combinations is less than $2 \times 3 \times 2 \times 2 \times 3 \times 4 \times 2 \times 3 = 864$.

Note also that some of the combinations can be equivalent for some query sets; in particular, in-query frequency has no effect for the queries based on titles, in which query terms are not repeated. Others are mathematically equivalent, thus giving ties in the rankings shown later.

relevance judgements are available. Disk 2 contains four document collections, and we partitioned these to make two experimental collections: a collection of newspaper articles (`ap2` and `wsj2`); and a collection of non-newspaper text (`fr2` and `ziff2`). Table 7 gives some statistics for these two collections.

Queries Of the various TREC topics, those numbered from 1 to 300 (excluding 201) have relevance judgements against the data of disk 2. Of these, topics 51–200 have been used extensively in previous TREC-related work as long queries; and they can also be used in two shorter forms—by taking the title section only, and by taking the narrative section only. These are what we used in our experiments. Table 8 shows an example of each of the three categories of query, namely those derived from TREC topic 200. Table 9 lists some information about the query sets.

Measurement of retrieval effectiveness For each query, each collection, and each similarity measure the top 1,000 ranked documents were identified, and a postprocessing program (`trec_eval`, available from `ftp://cs.cornell.edu`) used to obtain statistics about the relative performance of that mechanism. Of the many effectiveness measures reported by that program, those used in our experiments were the 11-point recall-precision average at 1,000 retrieved documents (averaging the precision attained at 0%, 10%, 20%, ..., 100% recall levels); the precision at 20 retrieved documents; and (after modifying `trec_eval`) the average value of $1/r_1$, where r_1 is the rank of the first relevant document returned. The use of three different effectiveness metrics added a final dimension to our burgeoning collection of statistics.

Results Tables 10, 11, and 12 show a partial summary of the data that was collected. Each of the six sections in each of the tables shows one of the six experimental domains: title, narrative, or full queries; and one of the two document collections. Inside each section there are 15 rows of data. The first of these, marked `ZZ-ZZZ-ZZ`, is discussed below. The next 10 rows show, for that combination of query set and collection, the best 10 of the 720 similarity measures that were explored, where best in Table 10 is judged by average 11-point recall-precision average over the query set, best in Table 11 is assessed by considering the average precision-at-20 value for the query set, and best in Table 12 is scored by average reciprocal rank of the first relevant document retrieved. The final four rows in each section of the tables show in snapshot fashion the extent to which the effectiveness degrades further down the ordering.

Each section of the three tables also contains a row marked `ZZ-ZZZ-ZZZ`. The score associated with this measure for each query is the best score achieved by any of the tested measures for that query. That is, each of the `ZZ-ZZZ-ZZZ` scores reflects the average (over the query set) of the best (over the 720 formulations) performing heuristic, and so represents the score that a clairvoyant user of the system—one able to decide in advance what formulation is best suited for each query—would obtain. The `ZZ-ZZZ-ZZ` measure gives an indication of the "goodness" (or otherwise) of the performance of the non-clairvoyant (and hence practical) mechanisms listed in the three tables.

It is obvious from these results that there is no measure that is a clear winner. There is little overlap between the successful measures in the eighteen cases—not only are different measures best for the different queries and data sets, but recall-precision yields somewhat different results to precision-at-20 and top-rank. For example, of the 20 factors we tested, just one of them (formulation E for the relative document frequency, Table 4) does not appear in a "top 10" measure listed in the three tables, and 87 different combinations appear amongst these "top 10" tables.

Furthermore, it would be incorrect to try and draw conclusions such as that certain measures work well in certain domains—as the results for the synthetic `ZZ-ZZZ-ZZ` measure show, none of the measures is particularly good for either a certain data set, a certain style of query, or a certain effectiveness metric. Particular measures do seem to work well for individual queries, but it is likely to be extremely difficult to recognise in advance which combinations will work in which cases.

These results not only make it difficult to identify successful measures, but also to identify and explain successful components. For example, pivoting is valuable some of the time but not all of the time; and it is not possible to categorically state that any particular weighting scheme is valuable. However, some trends do emerge. Not surprisingly, document length is important—combining function A is highly ranked only with relative term frequency F, the only relative term frequency to incorporate document length; and it seems as if the `A*-*F` formulations are better for the short queries (except when effectiveness is measured

	ap2wsj2	fr2ziff2
Size (megabytes)	479.3	384.8
Documents	154,439	76,780
Average document length (kilobytes)	3.2	5.1
Maximum document length (kilobytes)	133.0	1,836.5

Table 7: Statistics of document collections

Query set	Example query
Title	impact foreign textile imports textile industry
Narrative	impact positive negative qualitative may include expansion shrinkage markets manufacturing volume influence methods strategies textile industry textile industry includes production purchase raw materials basic processing techniques dyeing spinning knitting weaving manufacture marketing finished goods research textile field
Full	impact foreign textile imports textile industry foreign textiles textile products influenced impacted textile industry qualitative shrinkage markets manufacturing volume influence methods strategies textile industry textile industry production purchase raw materials basic processing techniques dyeing spinning knitting weaving manufacture marketing finished goods research textile field

Table 8: Examples of the three query sets

Collection	Query Set		
	Title	Narrative	Full
	Number		
ap2wsj2	150	150	150
fr2ziff2	101	101	101
	Average terms		
ap2wsj2	78.6	31.8	3.8
fr2ziff2	79.0	33.5	3.7
	Average answers		
ap2wsj2	90.4	90.4	90.4
fr2ziff2	17.7	17.7	17.7

Table 9: Statistics of query sets

Collection	Title		Narrative		Full	
ap2wsj2	*Q*	11-pt	*Q*	11-pt	*Q*	11-pt
	ZZ-ZZZ-ZZZ	0.312	ZZ-ZZZ-ZZZ	0.342	ZZ-ZZZ-ZZZ	0.426
1	AI-AFD-BCA	0.265	BI-BCK-BCA	0.288	BI-ACI-BCA	0.362
2	AI-AFD-BEA	0.265	BI-BCI-BCA	0.286	BB-ACI-BCA	0.362
3	AI-BFD-ACA	0.265	BI-ACI-BCA	0.282	BD-ACI-BCA	0.362
4	AI-BFD-AEA	0.265	BB-BCK-BCA	0.280	BI-BCI-BCA	0.356
5	AI-AFD-BAA	0.265	BI-BCD-BCA	0.280	BI-BCK-BCA	0.355
6	AI-BFD-AAA	0.265	BD-BCK-BCA	0.280	BB-BCI-BCA	0.353
7	AI-AFK-BCA	0.263	BB-BCI-BCA	0.279	BD-BCI-BCA	0.353
8	AI-BFK-ACA	0.263	BD-BCI-BCA	0.279	BB-BCK-BCA	0.350
9	AI-AFK-BEA	0.262	BD-ACI-BCA	0.277	BD-BCK-BCA	0.350
10	AI-BFK-AEA	0.262	BB-ACI-BCA	0.277	BI-ACD-BCA	0.350
100	AI-ACK-BCA	0.246	AI-BCB-BCA	0.232	BB-BEK-ACA	0.298
200	BB-BCI-AEA	0.240	AB-BCI-BEA	0.204	BB-BEK-AEA	0.270
400	BB-ACI-AEA	0.206	AB-BCI-ACA	0.161	BD-BFD-AAA	0.226
720	BI-AED-AAA	0.074	AI-ACK-AAA	0.046	AI-ACK-AAA	0.077
fr2ziff2	*Q*	11-pt	*Q*	11-pt	*Q*	11-pt
	ZZ-ZZZ-ZZZ	0.349	ZZ-ZZZ-ZZZ	0.415	ZZ-ZZZ-ZZZ	0.472
1	AB-BFD-BAA	0.231	BB-BCI-BCA	0.241	BD-BFK-BCA	0.294
2	AB-BFD-BCA	0.231	BD-BCI-BCA	0.241	BB-BFK-BCA	0.293
3	AB-BFD-BEA	0.231	BD-BFK-BCA	0.237	BB-BCI-BCA	0.281
4	AD-BFD-BAA	0.230	BB-BFK-BCA	0.236	BD-BCI-BCA	0.281
5	AD-BFD-BCA	0.230	BD-BCD-BCA	0.232	BI-BFK-BCA	0.275
6	AD-BFD-BEA	0.230	BB-BCD-BCA	0.232	BD-ACI-BCA	0.274
7	AB-BFK-BAA	0.228	BI-BCI-BCA	0.230	BB-ACI-BCA	0.274
8	AD-BFK-BAA	0.228	BB-BCI-BEA	0.229	BI-BCI-BCA	0.270
9	AB-BFK-BCA	0.228	BD-BCI-BEA	0.229	BD-BCD-BCA	0.266
10	AB-BFK-BEA	0.228	BD-BFK-BEA	0.218	BB-BCD-BCA	0.266
100	AD-BEB-BCA	0.194	BI-BEI-BEA	0.164	BI-BCB-ACA	0.201
200	AI-AEB-BCA	0.190	BI-BCI-AEA	0.133	BD-AFD-BAA	0.158
400	AI-BCD-AAA	0.161	AD-BED-AAA	0.091	AD-BEB-AEA	0.103
720	BI-AED-AEA	0.054	BI-AED-AAA	0.033	BI-AED-AAA	0.049

Table 10: Eleven-point recall-precision average at 1,000 documents returned

by precision and the collection is fr2ziff2), while B*-[AB] formulations handle the long queries better. But these two observations are about all that can be claimed.

In other experiments, not detailed here, we tested apparently trivial variants to the weighting schemes such as varying the base of the logarithm in relative term frequency C. These changes (from say $\log_2$ to $\log_e$) could have substantial effect on recall-precision. While testing mg we have at times been puzzled by our inability to exactly reproduce the results obtained by other researchers. It is now apparent that large differences in results can easily be the consequence of minor variations to the similarity measures such as the base of logarithms, and whether, as in another case we encountered, the "+1" addition takes place before or after the logarithms are taken.

Tables 13, 14, and 15 further illustrate this volatility. By normalising each recall-precision, precision-at-20, and top-rank score to a percentage of the ZZ-ZZZ-ZZZ score for that experimental domain, a set of "same unit" quantities can be calculated and further combined in different ways. Table 13 averages these scores over collections and effectiveness metrics, to show any trend that might be a result of the different characteristics of the three query sets; Table 14 averages the raw scores over collections and query sets, to identify any influences that can be attributed to the use of different effectiveness metrics; and Table 15 completes the third leg of the analysis, showing combined scores broken down by collection. In any of these three tables similarity formulations that work consistently well for that combination of parameters should appear near the top of a list of overall percentage, while those that are variable in their behaviour will

Collection	Title		Narrative		Full	
ap2wsj2	Q	pr@20	Q	pr@20	Q	pr@20
	ZZ-ZZZ-ZZZ	0.505	ZZ-ZZZ-ZZZ	0.556	ZZ-ZZZ-ZZZ	0.636
1	AB-AFD-BAA	0.405	BI-ACI-BCA	0.448	BI-ACI-BCA	0.521
2	AB-AFD-BEA	0.405	BB-ACI-BCA	0.447	BB-ACI-BCA	0.519
3	AB-BFD-AAA	0.405	BD-ACI-BCA	0.447	BD-ACI-BCA	0.519
4	AB-BFD-AEA	0.405	BI-ACK-BCA	0.436	BI-ACD-BCA	0.514
5	AI-AFK-BAA	0.405	BI-BCK-ACA	0.436	BI-BCD-ACA	0.514
6	AI-AFK-BCA	0.405	BB-ACK-BCA	0.432	BB-ACD-BCA	0.511
7	AI-AFK-BEA	0.405	BB-BCK-ACA	0.432	BB-BCD-ACA	0.511
8	AI-BFK-AAA	0.405	BD-ACK-BCA	0.432	BD-ACD-BCA	0.511
9	AI-BFK-ACA	0.405	BD-BCK-ACA	0.432	BD-BCD-ACA	0.511
10	AI-BFK-AEA	0.405	BI-ACI-BEA	0.432	BB-ACK-BCA	0.509
100	AD-BCK-AAA	0.400	AD-BFD-BCA	0.382	BD-BCB-AEA	0.459
200	AB-BCB-BCA	0.377	BB-BFD-AEA	0.349	BI-BCI-AAA	0.430
400	AI-AEI-AEA	0.342	BB-BEB-BEA	0.297	BI-BEI-AAA	0.379
720	BI-AED-AEA	0.143	AI-ACK-AAA	0.134	AI-ACK-AAA	0.186
fr2ziff2	Q	pr@20	Q	pr@20	Q	pr@20
	ZZ-ZZZ-ZZZ	0.195	ZZ-ZZZ-ZZZ	0.218	ZZ-ZZZ-ZZZ	0.256
1	BD-ACK-BAA	0.139	BB-BCI-BCA	0.150	BB-ACB-BCA	0.184
2	BD-ACK-BCA	0.139	BD-BCI-BCA	0.150	BD-ACB-BCA	0.184
3	BD-ACK-BEA	0.139	BB-BCI-BEA	0.148	BB-BFK-BCA	0.180
4	BD-BCK-AAA	0.139	BD-BCI-BEA	0.148	BD-BFK-BCA	0.180
5	BD-BCK-ACA	0.139	BB-BFK-BCA	0.146	BD-BCI-BCA	0.177
6	BD-BCK-AEA	0.139	BD-BFK-BCA	0.145	BB-BCI-BCA	0.176
7	BB-ACK-BAA	0.139	BI-BCI-BCA	0.143	BI-BCI-BCA	0.174
8	BB-ACK-BCA	0.139	BB-BCB-BEA	0.140	BB-ACB-BEA	0.170
9	BB-ACK-BEA	0.139	BB-ACB-BCA	0.140	BD-ACB-BEA	0.170
10	BB-BCK-AAA	0.139	BB-BCD-BCA	0.140	BB-BCD-BCA	0.169
100	AB-AFD-BCA	0.125	BI-ACI-BCA	0.108	BB-BFD-BAA	0.142
200	AI-AEB-BCA	0.118	BB-AEI-BAA	0.088	BB-AEI-BAA	0.117
400	AB-ACD-BAA	0.093	AD-AEI-BCA	0.049	AB-BEB-ACA	0.060
720	BI-AED-AEA	0.048	AI-ACK-AAA	0.031	AI-ACK-AAA	0.035

Table 11: Precision at 20 documents returned

appear lower.

The most striking feature of the three tables is again the poor performance. Even if we were lucky enough to select the right "best" measure for a combination of collection, effectiveness metric, and query set, that best mechanism would still only do roughly two thirds as well as if we could somehow include as a further parameter the actual query to be processed.

Finally, Table 16 averages the relative scores over all factors, to arrive at a single ranking of similarity functions. Given the lack of consistency in the individual experiments, the fairness of giving a single score to each mechanism is, however, debatable. The BB-ACB-BAA mechanism illustrated in Table 6 is ranked 88th of the 720 mechanisms, with a score of 57.1%.

Statistical significance One obvious possibility is that the volatility of the results is a consequence not of fundamentally altered behaviour, but of random fluctuation. To be sure that this was not the case, we applied the correlated t test [Graziano and Raulin 1993, p. 373] to the individual query results for pairs of measures. The results were unequivocal. For example, consider measures BI-BCI-BCA and BB-BCI-BCA on the full queries and the ap2wsj2 collection. These methods both score highly (Table 10), differ in just one factor, and differ in performance by just 0.003 averaged over the 150 queries. Yet the first measure outperforms the second on 99 queries, while the second is better on only 49; and this difference is enough to give a t value of 2.58, which is equal to the 99% confidence for this many observations. That is, the

Collection	Title		Narrative		Full	
ap2wsj2	Q	rank	Q	rank	Q	rank
	ZZ-ZZZ-ZZZ	0.840	ZZ-ZZZ-ZZZ	0.936	ZZ-ZZZ-ZZZ	0.966
1	AB-AFK-AAA	0.644	BI-ACK-BCA	0.676	BD-ACI-BCA	0.746
2	AB-AFK-ACA	0.644	BI-BCK-ACA	0.676	BB-ACI-BCA	0.746
3	AB-AFK-AEA	0.644	BI-ACI-BEA	0.670	BI-ACI-BCA	0.739
4	AD-AFK-AAA	0.644	BD-ACI-BCA	0.669	BB-ACK-BCA	0.734
5	AD-AFK-ACA	0.644	BB-ACI-BCA	0.669	BB-BCK-ACA	0.734
6	AD-AFK-AEA	0.644	BB-ACK-BCA	0.667	BD-ACK-BCA	0.734
7	AI-AFK-AAA	0.644	BB-BCK-ACA	0.667	BD-BCK-ACA	0.734
8	AI-AFK-ACA	0.644	BD-ACK-BCA	0.667	BB-ACI-BEA	0.733
9	AI-AFK-AEA	0.644	BD-BCK-ACA	0.667	BD-ACI-BEA	0.733
10	AI-ACB-BAA	0.642	BI-ACI-BCA	0.666	BI-ACI-BEA	0.732
100	AI-BCD-BAA	0.622	BD-BCK-AAA	0.600	BI-AEI-BEA	0.685
200	BD-BCK-AEA	0.590	BI-BEK-BCA	0.568	BB-AEK-ACA	0.653
400	AD-BEB-BAA	0.533	BI-AFD-ACA	0.506	AB-BED-BEA	0.591
720	BI-AED-AAA	0.308	AI-ACK-AAA	0.257	AI-ACK-AAA	0.338
fr2ziff2	Q	rank	Q	rank	Q	rank
	ZZ-ZZZ-ZZZ	0.662	ZZ-ZZZ-ZZZ	0.758	ZZ-ZZZ-ZZZ	0.801
1	AD-AFD-BAA	0.400	BB-BCI-BCA	0.467	BI-BCI-BCA	0.490
2	AD-AFD-BEA	0.400	BD-BCI-BCA	0.467	BB-ACB-BCA	0.489
3	AD-BFD-AAA	0.400	BB-BCI-BEA	0.449	BD-ACB-BCA	0.489
4	AD-BFD-AEA	0.400	BD-BCI-BEA	0.448	BI-BFK-BCA	0.488
5	AB-AFD-BAA	0.400	BI-BCI-BCA	0.447	BB-BCI-BCA	0.483
6	AB-AFD-BEA	0.400	BD-BFK-BCA	0.437	BD-BCI-BCA	0.482
7	AB-BFD-AAA	0.400	BB-BFK-BCA	0.436	BB-BFK-BCA	0.482
8	AB-BFD-AEA	0.400	BD-BCD-BCA	0.424	BD-BFK-BCA	0.482
9	AD-AFD-BCA	0.399	BB-BCD-BCA	0.423	BB-ACB-BEA	0.476
10	AD-BFD-ACA	0.399	BD-ACI-BCA	0.419	BD-ACB-BEA	0.476
100	AI-BEB-BEA	0.377	BD-AFK-BCA	0.365	BB-BFK-BAA	0.407
200	AD-BEB-BEA	0.361	BB-BEB-AEA	0.304	BI-BCK-BEA	0.336
400	BD-BCD-BCA	0.317	AD-AFD-BEA	0.203	AB-AEB-BCA	0.224
720	BI-AED-AAA	0.121	BI-ACK-AAA	0.123	BI-ACK-AAA	0.135

Table 12: Average (reciprocal) rank of first relevant document

superiority of BI-BCI-BCA over BB-BCI-BCA is almost certainly significant on collection ap2wsj2 and the full queries with effectiveness measured by 11-point recall-precision average at 1,000 documents retrieved.

Table 17 reports six more pairwise tests, comparing on each of the three types of query the three "best" mechanisms (for recall-precision on ap2wsj2) in a "home and away" competition. The first of the three sections in the table shows that AI-AFD-BCA is clearly better than either BI-BCK-BCA or BI-ACI-BCA when the queries are short; while the remaining two sections show that AI-AFD-BCA should very definitely *not* be used for long queries on this collection—in each case the high calculated t values indicate that random chance has nothing to do with the relative performance of the various methods.

Similar confidence scores are calculated when the the fr2ziff2 best methods are compared with their partners in ap2wsj2, again using recall-precision as the effectiveness metric. For example, using the title queries, AI-AFD-BCA is better than AB-BFD-BAA with a confidence of $t = 2.53$ in ap2wsj2, and in fr2ziff2 AB-BFD-BAA is better than AI-AFD-BCA with a confidence of $t = 2.40$, which both exceed the 95% confidence limit of 2.0. Changing from one effectiveness metric to another gives similar chaotic results. Using ap2wsj2 and the title queries, an 11-point recall-precision comparison between AI-AFD-BCA and AB-AFD-BAA (which is the best according to the precision metric) gives a score of $t = 3.57$.

That is, the choice of retrieval mechanism is governed not just by the characteristics of the queries being processed, but also by the characteristics of the data that is being handled (even for the same query

	Title		Narrative		Full	
	ZZ-ZZZ-ZZZ	100.000	ZZ-ZZZ-ZZZ	100.000	ZZ-ZZZ-ZZZ	100.000
1	AI-BFD-BAA	70.926	BB-BCI-BCA	68.153	BD-ACI-BCA	70.916
2	AI-BFD-BEA	70.852	BD-BCI-BCA	68.129	BB-ACI-BCA	70.877
3	AD-AFD-BEA	70.821	BI-BCI-BCA	67.398	BD-BCI-BCA	70.425
4	AD-BFD-AEA	70.821	BD-ACI-BCA	66.037	BD-ACB-BCA	70.418
5	AB-AFD-BEA	70.814	BB-ACI-BCA	65.992	BB-BCI-BCA	70.401
6	AB-BFD-AEA	70.814	BB-BCI-BEA	65.814	BB-ACB-BCA	70.397
7	AD-AFD-BAA	70.811	BD-BCI-BEA	65.790	BI-BCI-BCA	70.162
8	AD-BFD-AAA	70.811	BD-BCD-BCA	64.895	BD-BFK-BCA	68.480
9	AB-AFD-BAA	70.803	BB-BCD-BCA	64.884	BB-BFK-BCA	68.479
10	AB-BFD-AAA	70.803	BD-ACB-BCA	64.384	BI-ACI-BCA	68.182
100	AB-BCB-BEA	65.516	BD-BEK-ACA	54.835	BI-AEB-BCA	58.996
200	AB-AEB-BEA	63.887	BI-BCB-AAA	48.594	BB-BEB-ACA	52.762
400	AD-AEI-AAA	58.685	AD-BEB-BAA	37.517	BD-AEK-AEA	42.283
720	BI-AED-AAA	24.498	AI-ACK-AAA	19.428	AI-ACK-AAA	22.099

Table 13: Mechanisms with best overall performance when grouped by query type

	Eleven-point average		Precision at 20		Rank of first relevant	
	ZZ-ZZZ-ZZZ	100.000	ZZ-ZZZ-ZZZ	100.000	ZZ-ZZZ-ZZZ	100.000
1	BD-ACI-BCA	67.873	BD-ACI-BCA	71.180	BD-ACI-BCA	64.507
2	BB-ACI-BCA	67.828	BB-ACI-BCA	71.156	BB-ACI-BCA	64.490
3	BB-BCI-BCA	67.167	BD-BCI-BCA	70.358	BI-ACI-BCA	62.260
4	BD-BCI-BCA	67.157	BB-BCI-BCA	70.357	BD-ACI-BEA	61.588
5	BI-BCI-BCA	67.093	BI-BCI-BCA	70.044	BB-ACI-BEA	61.566
6	BD-BCK-BCA	66.501	BI-ACI-BCA	69.194	BB-BCI-ACA	61.535
7	BB-BCK-BCA	66.482	BB-BCI-BEA	68.735	BD-BCI-ACA	61.513
8	BD-BCD-BCA	65.904	BD-BCI-BEA	68.725	BB-BCI-BCA	61.426
9	BB-BCD-BCA	65.899	BD-ACI-BEA	68.266	BD-BCI-BCA	61.412
10	BI-ACI-BCA	65.629	BB-ACI-BEA	68.247	BI-BCI-BCA	61.207
100	BD-BCK-AEA	54.078	BB-BCI-AAA	59.352	BB-BCB-BCA	54.659
200	BB-BCD-AAA	49.760	BD-AEB-BEA	54.232	BD-BFD-ACA	52.312
400	AI-AEB-BEA	41.814	BI-BED-ACA	47.367	AI-BEB-ACA	48.346
720	BI-AED-AAA	18.740	BI-AED-AAA	28.173	BI-AED-AAA	32.139

Table 14: Mechanisms with best overall performance when grouped by effectiveness metric

characteristics), by the effectiveness metric that is being used (even for the same query characteristics and same database), and by the terms present in those queries (even for the same query characteristics and same database and same effectiveness metric).

4 Conclusions

We commenced this investigation with several aims. First, we wished to enumerate previously articulated similarity functions and cast them into a uniform framework with systematic nomenclature. This alone we felt to be worthwhile, and we believe that others will find our notation and formulations useful and worth adoption.

Our second goal was to experiment with these measures, explore the space they define, and identify good measures and good components. We did not achieve this goal—indeed, we now suspect that it is unattainable. The measures do not form a space that can be explored in any meaningful way, other than by exhaustion. Even restricting ourselves to a subspace including several measures that were thought to work well, we not only failed to find any particular measure that really stood out but discovered that no

	ap2wsj2		fr2ziff2	
	ZZ-ZZZ-ZZZ	100.000	ZZ-ZZZ-ZZZ	100.000
1	BD-ACI-BCA	78.273	BD-BCI-BCA	59.279
2	BB-ACI-BCA	78.268	BB-BCI-BCA	59.268
3	BI-ACI-BCA	78.258	BI-BCI-BCA	57.953
4	BI-ACK-BCA	77.388	BB-BCD-BCA	57.457
5	BI-BCK-ACA	77.388	BD-BFK-BCA	57.453
6	BD-ACK-BCA	77.074	BB-BFK-BCA	57.449
7	BD-BCK-ACA	77.074	BD-ACI-BCA	57.433
8	BB-ACK-BCA	77.071	BD-BCD-BCA	57.419
9	BB-BCK-ACA	77.071	BB-ACI-BCA	57.382
10	BI-ACI-BEA	75.996	BB-BCI-BEA	57.269
100	AB-BCK-BCA	69.376	BI-BCD-BAA	46.080
200	BD-AEI-BCA	64.005	BD-ACK-BCA	40.729
400	AD-AEI-BCA	57.391	AB-AEI-BAA	34.829
720	BI-AED-AAA	34.827	BI-AED-AAA	17.874

Table 15: Mechanisms with best overall performance when grouped by collection

	Overall	
	ZZ-ZZZ-ZZZ	100.000
1	BD-ACI-BCA	67.853
2	BB-ACI-BCA	67.825
3	BB-BCI-BCA	66.317
4	BD-BCI-BCA	66.309
5	BI-BCI-BCA	66.114
6	BI-ACI-BCA	65.694
7	BB-BCI-BEA	64.189
8	BD-ACI-BEA	64.185
9	BB-ACI-BEA	64.165
10	BD-BCI-BEA	64.165
100	BB-BFK-BAA	55.947
200	BB-BFD-BEA	52.169
400	BD-BED-BCA	46.260
720	BI-AED-AAA	26.351

Table 16: Mechanisms with best overall performance

measure consistently worked well across all of the queries in a query set. Both average-case performance and individual-case performance is poor for even the best measures overall. Likewise, no component or weighting scheme was shown to be consistently valuable across all of the experimental domains; that is, success in one domain was a poor predictor for success in another. Moreover, variations such as choice of base for a logarithm could have as profound an effect as more principled modifications; it is thus difficult to assess whether "improvements" are the result of better understanding of the problem of information retrieval, or are simply the chance peaks that arise in complex domains.

It is, however, clear that better performance can be obtained—by choosing a similarity measure to suit each query on an individual basis. But it seems implausible to suppose that a mechanism for making such a choice could be found, or that the weights for a grand combination-of-evidence mechanism could be sensibly chosen. When evaluating a query, the type of data, the type of query, the query itself, the evaluation metric, and as far as we know the type of answer, all matter. Looked at another way, the variability is sufficiently great that no single method attains better than about two thirds of what we now know can be attained by an ideal mechanism.

This work is not complete, but we cannot see any clear route forward that would allow us to bring it to a satisfactory conclusion. Indeed, in many ways the contradictory and confusing results we have achieved

Correlated t test results				
Query set	Title		Title	
Formulation	AI-AFD-BCA	BI-BCK-BCA	AI-AFD-BCA	BI-ACI-BCA
Mean value	0.265	0.245	0.265	0.247
Wins	96	54	94	56
Calculated t	3.4		3.5	
Query set	Narrative		Narrative	
Formulation	BI-BCK-BCA	AI-AFD-BCA	BI-BCK-BCA	BI-ACI-BCA
Mean value	0.288	0.211	0.288	0.282
Wins	118	31	73	76
Calculated t	9.2		1.3	
Query set	Full		Full	
Formulation	BI-ACI-BCA	AI-AFD-BCA	BI-ACI-BCA	BI-BCK-BCA
Mean value	0.362	0.296	0.362	0.355
Wins	123	27	88	62
Calculated t	9.8		1.4	

Table 17: Significance results using the correlated t test. All entries refer to eleven-point recall-precision average (Table 10) and the `ap2wsj2` collection

discourage us from hoping that a "silver bullet" for information retrieval can ever be found. We welcome any suggestions that you may have that might help us discern the patterns in this apparently chaotic behaviour. In the meantime, when in doubt, use `BD-ACI-BCA`.

Data The retrieval effectiveness results summarised in Tables 10, 11, and 12 are available in full at `http://www.cs.mu.oz.au/~alistair/exploring/`. We hope that other researchers will avail themselves of this resource.

Acknowledgements Tim Shimmin and Owen de Kretser undertook the programming work involved for this paper, and we thank them for their patience in dealing with the many conflicting requirements we placed on them. We are also grateful to Ross Wilkinson for his valuable advice. This work was supported by the Australian Research Council and the Key Centre for Knowledge-Based Systems.

References

BELL, T., MOFFAT, A., WITTEN, I., AND ZOBEL, J. 1995. The MG retrieval system: compressing for space and speed. *Communications of the ACM 38*, 4 (April), 41–42.

FRAKES, W. AND BAEZA-YATES, R., Eds. 1992. *Information Retrieval: Data Structures and Algorithms.* Prentice-Hall, Englewood Cliffs, New Jersey.

GRAZIANO, A. AND RAULIN, M. 1993. *Research Methods: A Process of Enquiry.* Harper Collins, New York.

HARMAN, D. 1995. Overview of the second text retrieval conference (TREC-2). *Information Processing & Management 31*, 3 (May), 271–289.

KORFHAGE, R. 1997. *Information Storage and Retrieval.* Wiley, New York.

MG-software. 1995. MG public domain software for indexing and retrieving text, including tools for compressing text, bilevel images, grayscale images, and textual images. Available from `ftp://munnari.oz.au/pub/mg` and from `http://www.mds.rmit.edu.au/mg/`.

MOFFAT, A. AND ZOBEL, J. 1994. Compression and fast indexing for multi-gigabyte text databases. *Australian Computer Journal 26*, 1 (February), 1–9.

OZKARAHAN, E. 1986. *Database Machines and Database Management.* Prentice-Hall.

ROBERTSON, S., WALKER, S., AND HANCOCK-BEAULIEU, M. 1995. Large test collection experiences on an operational, interactive system: Okapi at TREC. *Information Processing & Management 31*, 3 (May), 345–360.

SALTON, G. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley, Reading, MA.

SALTON, G. AND BUCKLEY, C. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management 24*, 5, 513–523.

SALTON, G. AND MCGILL, M. 1983. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York.

SINGHAL, A., BUCKLEY, C., AND MITRA, M. 1996. Pivoted document length normalization. In H.-P. Frei, D. Harman, P. Schäuble, and R. Wilkinson, Eds., *Proceedings of the Nineteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval.* Zurich, Switzerland, pp. 21–29.

VAN RIJSBERGEN, C. 1979. *Information Retrieval* (second ed.). Butterworths.

WITTEN, I., MOFFAT, A., AND BELL, T. 1994. *Managing Gigabytes: Compressing and Indexing Documents and Images.* Van Nostrand Reinhold, New York.

ZOBEL, J. 1998. How reliable are the results of large-scale information retrieval experiments? Technical Report 98-1, Department of Computer Science, RMIT, Melbourne.