Term position ranking: some new test results

E Michael Keen

Department of Information and Library Studies
University College of Wales Aberystwyth, UK

Abstract

Presents seven sets of laboratory results testing variables in term position ranking which produce a phrase effect by weighting the distance between proximate terms. Results of the 73 tests conducted by this project are included, covering variant term position algorithms, sentence boundaries, stopword counting, every pairs testing, field selection, and combinations of algorithm including collection frequency, record frequency and searcher weighted. The discussion includes the results of tests by Fagan and by Croft, the need for term stemming, proximity as a precision device, comparisons with Boolean, and the quality of test collections.

Introduction

Some laboratory evaluation tests which started at Aberystwyth in 1990 have concentrated on algorithms for query-record matching based on term position or proximity. In this non-Boolean search method records are ranked in order of decreasing match with a query by a weighted score reflecting both the number of terms that match and their proximity to one another in the fields and sentences of the records. Thus the closer the terms are together the higher the match: the way this 'pairs distance' score is computed is illustrated in the Test 2 section which follows. The first test results were presented in Ref. 1 and were described as the automated analogy of the conventional Boolean searcher's tactic of specifying field position or word proximity to narrow the search and improve Precision. Compared with the simplest output ranking method of counting the number of matching terms (quorum match) the best of the term position algorithms improved performance by between 12% and 18% in Precision depending on the level of Recall.

15th Ann Int'l SIGIR '92/Denmark-6/92
© 1992 ACM 0-89791-524-0/92/0006/0066...$1.50

Refs. 2-5 reported further tests which incorporated three other well known techniques of output ranking into the comparisons, namely, inverse document frequency, term frequency and query terms weighted by the searcher. It was found that pairs distance was usually better than any of these methods, and when many combination methods were tried the presence of the pairs distance algorithm nearly always improved performance more than the other methods (see Ref. 5 especially). At the time of writing this paper (December 1991) 73 test results have been obtained, 36 of which have been published. It is to the remaining 37 tests that this paper will turn as many of them explore some of the myriad of potentially important variables in term position matching and other output ranking algorithms.

Table 1 gives a list of the 73 tests made on one test collection of 6004 records and listed in order of a measure described as Merit. This measure is the arithmetic mean Precision Ratio computed from the three Precision Ratios which each of the 35 queries achieves at the three standard Recall Ratio levels of 25%, 50% and 75%. It is the measure used by Salton in presenting his many hundreds of tests of term weighting algorithms (Ref. 6). The 73 results can be categorised as follows:

5 results, including quorum or coordination level matching, are performance benchmarks

48 results test term position algorithms (20 in combination with other algorithms)

28 results test collection frequency, otherwise known as inverse document frequency (23 in combination)

15 results test record frequency, otherwise known as term frequency (14 in combination)

16 results test searcher weighted, described in Ref. 5 (15 in combination).

It can be seen, therefore, that the 68 non-benchmark tests can be divided into 34 which test one single match algorithm, and 34 which test combinations of algorithm.

The final column of Table 1 lists the variables which are the subject of the tests reported here. Published comparisons will not be repeated unless a new candidate can be added. The work which looked at stemming or suffixing published in Ref. 2, and the comparisons with Boolean searching

% Merit	Rank	System	Refs & variables
100	1	Best Possible	1,3,4,5 benchmark
55.8	2	Pairs & Collectn & Record & Searcher	5
54.6	3	Pairs D & Collection & Record f	5
54.3	4	Pairs Distance & Collection Frequency	4,5
54.3	4	Pairs Distance & Searcher Weighted	5
54.3	4	Pairs Distance & Searcher Weighted	Pairs score/5
54	7	Pairs Distance & Collection Frequency	Equation (a)
53.8	8	Best Possible Within Quorum	1 benchmark
53.4	9	Pairs D & Collection F & Searcher W	5
53	10	Distance wn prox pairs & collection f	Equation (b)
52.9	11	Pairs D & Searcher W & Record F	5
52.7	12	Distance wn prox pairs & Collection f	Equation (a)
51.9	13	Proximate pairs & collection frequency	Equation (c)
51.6	14	Collection F & Record F & Searcher W	5
51.4	15	Proximate pairs & collection frequency	(c) singles posted
51.1	16	Collection & Record Frequency	3,4 (c)
51.1	16	Collection F & Record F & Searcher W	Equation (a)
51	18	Collection & Record Frequency	5
50.8	19	Distance within prox pairs	No sentences
50.3	20	Collection & Record Frequency	Equation (a)
49.4	21	Pairs Distance Product	5
48.7	22	Distance within (prox) pairs (algor 7)	1,3,9
48.4	23	Proximate pairs & searcher weighted	Equation (c)
48.3	24	Pairs Distance & Record Frequency	5
47.5	25	Collection F & Record F & Searcher W	Equation (c)
47.5	25	Distance within prox pairs (sum)	Stopwords uncounted
47.4	27	Total distance & collection frequency	Equation (c)
47.2	28	Distance within prox pairs (product)	Stopwords uncounted
47.1	29	Proximate pairs & collection frequency	Equation (a)
46.7	30	Searcher Weighted & Record Frequency	5
45.5	31	Proximate pairs stem wts	2 stem
44.8	32	Sentences totalled wq (algor 1)	1 wq
44.7	33	Pairs distance product	Within quorum
44.4	34	Proximate pairs (algor 4)	1,2,3
44.4	34	Total distance & collection frequency	Equation (a)
44.3	36	Distance within prox pairs	No sentences
43.8	37	Pairs distance sum	Proximity algorithm
43.7	38	Pairs distance sum wq	Within quorum
43.5	39	Dist within prox pairs wq (algor 7)	1 wq
43	40	Collection Frequency	2,3,4,5
43	40	Collection Frequency & Searcher Weighted	Equation (a)
42.9	42	Collection Frequency	Integer 1-6
42.7	43	Collection Frequency & Searcher Weighted	5
42.3	44	Proximate pairs wq (algor 4)	1 wq
42	45	Collection frequency s-stem	2 s-stem
41.3	46	Proximate pairs s-stem	2 s-stem
41.3	46	Proximate pairs	Every pair
41.2	48	Sentences totalled (algor 1)	1
41.2	48	Searcher Weighted & Record Frequency	Equation (a)
41.1	50	Collection frequency priority stems	2 priority stem
40.5	51	Mean distance & collection frequency	Equation (a)
39.8	52	Dist 1 within best sent wq (algor 6)	1 wq
39.8	52	Mean distance & collection frequency	Equation (c)
39.5	54	Total distance	Proximity algorithm
39.4	55	Collection Frequency & Searcher Weighted	Equation (c)
39.3	56	Searcher Weighted & Record Frequency	Equation (c)
38.6	57	Searcher Weighted	5
38.3	58	Best sentence wq (algor 2)	1 wq
35.4	59	Proximate pairs & record frequency	Equation (c)

Table 1 (first part)

% Merit	Rank	System	Refs & variables
34	60	Quorum stem weights	2 stem wts
34	60	Quorum, stem wq.	Stem wq
33.7	62	Quorum s-stem	2 s-stem
33.1	63	Record Frequency	3,5
32.9	64	Quorum	1,2,3,4,5,9
32.2	65	Sentences totalled wq	No DE field, wq
30.1	66	Mean distance	Proximity algorithm
29.7	67	Direct collection frequency	3,4 benchmark
29.2	68	Direct collection frequency	Integer 1-6
28	69	Quorum	No DE field
26.9	70	Best sentence (algor 2)	1
21	71	Sentences token	Proximity algorithm
19.5	72	Sentences totalled	No DE field
4.2	73	Random	4 benchmark

Table 1: The 73 test results as ranked by merit (mean precision ratio at three recall thresholds) using 35 queries and 6004 records from Library and Information Science Abstracts.

presented in Ref. 4, will not be presented here as no additional comparisons have been made. The new test results are now presented, divided into seven types of test.

Test 1: Benchmarks

The five benchmarks results are ranked 1, 8, 64, 67 and 73 in Table 1. In Table 2 the Merit performance measure for these five results is repeated along with nine other performance results, consisting of three 'Precision' measures at each of the three Recall thresholds. For Best possible the Output position measure is influenced by the number of relevant per query, which is 10.8 (mean), 7 (median) in this test collection. The Best possible within quorum was computed simply by placing the relevant at the front rank positions in each of the weakly ordered quorum match levels. Table 1 shows that six results, ranked 2 through 7, achieved a better performance than this benchmark. Quorum is really both a benchmark for these experiments and, of course, a system in its own right. Direct collection frequency weighted was described as 'perverse document frequency' in Ref 3 as it weights the frequent terms high and the infrequent low as a reverse benchmark to judge effectiveness of the normal collection frequency/idf theory: Collection frequency ranks 40, Merit 43.0%, which is significantly better than the perverse algorithm with its rank of 67 and Merit of 29.7%

The random benchmark placed the relevant items randomly amongst the total set of matched records, not the total collection, as the mean output position for 75% Recall indicates, which is 448.5, rather than 4503 which would be the rank if the whole collection were considered. It is doubtful whether this random benchmark is of any value: its result is so much lower than the worst system tested. Three of these benchmarks are included in the Precision/Recall graphs presented by this project in Ref.3, and four in Ref. 4.

Test 2: Term position algorithms, including proximity

In proposing term position algorithms, Ref. 1 offered a typology of seven variants of four kinds, as follows:

Matching sentences:
(1) All sentences
(2) Best sentence
Proximate term counts:
(3) Density
(4) Pairs
Proximate term order:
(5) (no variant identified)
Distance intervening
(6) Between extremes
(7) Between pairs

Algorithms (1), (2), (4), (6) and (7) have been tested and results appear in Table 1. Algorithm (7) nearly always performed best, and the following example (similar to Fig 1(c) Ref 1. page 7) shows how the matching between query and record is performed:

Query Terms: A B C D E F G H

Record example, showing matching query terms, non-matching content-bearing terms (*) and stopwords ($):

TITLE FIELD
* A * * C $ $ * E F $ * $ D * * $ *

ABSTRACT SENTENCE 1
$ * $ * E F $ C * * $ B $ * * $ * $ * $ * * * * *

ABSTRACT SENTENCE 2
* * $ * * * * * $ * $ * * * $ B C * * * $ * * *

ABSTRACT SENTENCE 3
* $ * $ $ B C * $ * * * $ * * * * * $ * $ * $ * *

ABSTRACT SENTENCE 4
$ * $ * * $ * $ $ * * * $ * * * * $ * * $ * * *

CONTROLLED HEADING
* F * * * *

CONTROLLED DESCRIPTORS
* * $ D G * $ H G H * * $ B G H C G * $ H * F

There are 20 candidate proximate pairs matches, listed here in the form 1st term/2nd term/number of intervening words:

AC2 CE3 EF0 FD3 EF0 FC1 CB3 BC0 BC0 DG0 GH2 HG0 GH0 HB4 BG0 GH0 HC0 CG0 GH2 HF1

The default method of counting intervening words included both content bearing ones and stopwords. The default method of dealing with these 20 candidates was to count each unique pair once, either way round, and choose the minimum distance count. Working from left to right, the result would be the following 13 pairs:

AC2 CE3 EF0 FD3 FC1 BC0 DG0 HG0 HB4 BG0 HC0 CG0 HF1

A simple linear inverse distance weight was then applied, with zero distance weighted 10, distance 1 weighted 9, and so on with distances of 9 or more weighted one. This gives the following score components:

AC8 CE7 EF10 FD7 FC9 BC10 DG10 HG10 HB6 BG10 HC10 CG10 HF9

Table 3 gives performance results for five sub-algorithms based on this default method of distance counting. The best result uses the pairs distance product: in this example, the number of pairs (13) times the mean of the pairs distances (116/13 = 8.9) resulting in a query/record match of 115.7 . A close second in merit is distance within the pairs: here the pairs score of 13 is the primary score, and within that the distance mean of 8.9 would be used to rank all records having a score of 13 by means of a simple function such as (13 x 10) + 8.9 = 138.9 . Rather lower performance is given by pairs distance sum alone, in this case, ie. 13 + 8.9 = 21.9 . Worse still is total distance, 116, and the poorest is the mean distance, 8.9 in the example.

These algorithms obviously are attempts to provide ranked output by recognising what other researchers call "statistical phrase matching". For example, Croft and others (Ref. 7) discuss five issues in this approach, and this research can be fitted in to their framework. Term position and proximity here use a distance or adjacency weighting that counts stopwords, rather than Croft's use of a fixed value, namely of 3 intervening non-stopwords. The "Aberystwyth" method is clearly suitable only for identification during the search, not for adding to the indexing.

Croft offers four phrase models and tests two of them. This research does seem to fit the two he did not test, his model (c) in which the "phrase is a dependancy relationship between components", and his model (d) ("belief in components dependent on belief in phrase") is represented by the combinations of weighting methods reported here as Test 7 and Table 5. The relationship between the components or concepts, here pairs of terms, being weighted by distance apart on a scale of ten does emphasise the fuzzy or probabilistic approach being advocated here: it is not important that every phrase is a valid one, or that every valid phrase is picked up. Indeed, some of the latter are not, as my own critique to follow shortly will show.

It is also a major feature of Fagan's comprehensive study of statistical phrases on the SMART system that his proximity distance values are fixed, though he did test four different values from unlimited distance down to very close proximity (Ref. 8). Fagan isolated six major parameters for statistical phrases, and conducted experiments varying five of them. The domain, or record window, is the first, with Fagan testing both document and sentence. For this work sentence was the default, as illustrated above, but Test 3 reports a test of the document domain.

Fagan's proximity used fixed distances and did not count stopwords, similar to Croft. Test 4 here reports a test in which stopwords were removed. Fagan's rules for phrase recognition used collection frequency (he calls it document frequency) considerations, with frequency thresholds and minima for the high frequency term in each pair and the low frequency one, called df head and df component respectively. The term position algorithms and proximity sub-algorithms already presented show our different approach. Fagan's pairs did not have to occur adjacently, and in fact every pair of terms in the records and queries were regarded as phrases if they occurred less than 90 times in the collection. Test 5 will report this approach using every pair but without any frequency limit. Test 7 will introduce frequency parameters of three different kinds, not as thresholds but as term weighting devices of equivalent status to be combined with proximity weights. There are a number of other differences comparedwith Fagan's approach, such as the use of selective weak stemming rather than his universal use of strong stems. His work calls attention to earlier experiments with statistical phrases, and includes some work on syntactical phrases as well, so his paper is a good source for a summary of these lines of research.

It is argued that our approach avoids arbitrary frequency threshold decisions, utilises nearly all available query/matching events, but by avoiding every possible pair gives a more realistically achievable algorithm suitable for run-time implementation. There are a few cases where a term that matches a query and record is not picked up by the pairs matching because the term is a singleton in a field, such as would be the case in the above example with term F in the controlled heading field were it not for the fact that in this case F occurs in other fields. Such missing matches could be added into the algorithm in some way, and in Test type 7 this was done in one run using a combination of methods.

The restriction to adjacent pairs without an intervening matching query term yet accepting a match where there are non-matching terms is open to question. In the example above, pair CE3 occurs in the title yet the Abstract Sentence 1 includes the pattern "E F $ C" in which E and C are separated by one query term (F) and a stopword: this would seem to be a case of at least EC2, and maybe EC1 if term F

Scheme	Merit	Rank	Low Recall Threshold 25%			Medium Recall Threshold 50%			High Recall Threshold 75%		
			Precision ratio	Output position		Precision ratio	Output position		Precision ratio	Output position	
	mn		mn	md	mn	mn	md	mn	mn	md	mn
Best possible	100%	1	100%	2	3.2	100%	4	5.7	100%	6	8.0
Best possible within quorum	53.8%	8	66.4%	3.5	10.1	52.3%	10	19.9	42.7%	26	48.4
Quorum	32.9%	64	45.4%	9	21.5	31.1%	28	44.3	22.0%	79	111.1
Direct Collection Frequency Weighted	29.7%	67	41.8%	12	38.1	29.4%	30	72.8	17.8%	87	176.7
Random	4.2%	73	4.2%	144	171.4	4.2%	206	297.2	4.2%	341	448.5

Table 2: Five benchmark performance results.

Pairs distance product	49.4%	21	62.4%	4	9.7	51.3%	14	26.7	34.5%	44	90.9
Distance within Proximate Pairs	48.7%	22	62.7%	5	9.3	49.7%	14	26.7	33.5%	51	88.4
Pairs distance sum	43.8%	37	58.1%	4	13.6	44.3%	19	45.0	28.9%	61	124.7
Total distance	39.5%	54	52.2%	5	33.1	41.3%	14	66.7	25.1%	59	135.7
Mean distance	30.1%	66	39.5%	8	39.1	29.6%	18	81.4	22.6%	81	157.5

Table 3: Results comparing five pairs distance algorithms.

Adjacent Proximate Pairs	44.4%	34	58.9%	6	11.1	45.6%	13	32.4	28.4%	51	101.2
Every pair	41.3%	46	54.1%	6	13.0	40.2%	18	33.7	29.8%	66	105.1

mn = mean md = median

Table 4: Results comparing adjacent and every proximate pair both with sentence boundaries.

were to be regarded as a good thing and not to be counted in the distance value. It would not be very easy to design an algorithm which is more economic than processing every possible pair yet finds, say, these selected cases where an adjacent pair distance can be bettered by allowing the intervening matching term. I suppose it could be done on a distance basis: with everything further apart than 9 words being given a weight of one, cases of every pair would be limited to those occurring only up to 9 words apart, or perhaps 5.

Turning to a final topic in term position algorithms, the "within quorum" strategy tested in Ref. 1 has been the subject of a total of 9 comparison tests as Table 1 reveals (this method uses quorum matching as the primary ranking device and adds a second method to rank within the quorum levels). The strength of quorum is its clarity of match for the searcher: the matching levels can be displayed, the actual terms highlighted, and some of the feeling of match control so beneficial to professional Boolean searchers can be preserved, and if a secondary device can push the relevant items towards the front of the list within a level then this may offer a desirable scheme. (Indeed, there may well be a place for ranking methods subordinate to a fairly conventional Boolean search to order the output, as proposed using proximity devices by the present writer in Ref 9, and by CAIRS using matches on specified fields Ref. 10) 7 of the 9 within quorum tests used term position methods: using distance algorithms, within quorum worsened the performance (rank 21 dropped to 33, 22 to 39 and 37 to 38), and using just pairs counts also (rank 34 dropped to 44). But the poorer performing sentences algorithm was improved by within quorum (ranks 48 became 32, 70 became 58 and 72 became 65). It must therefore be concluded that no means has yet been found of preserving quorum and so approaching the performance of the benchmark "best within quorum", result rank 8, by a quorum-based approach.

Test 3: No sentence or field boundaries

What Fagan calls document domain has been varied in two tests, both using pairs distance. Removing boundaries slightly improves pairs distance product from rank 21 to 19, and it provides the best Precision Ratio (65.6%) at 25% Recall of all results not using combination methods. Boundary removal has a similarly slight opposite effect using distance within proximate pairs, as the rank falls from 22 to 36, but the merit measure difference is only 2.5%.

This result does further suggest that across boundary matches are not harmful to performance and may sometimes be helpful. They would remove altogether the cases of single matching terms not recognised as pairs, so together with some possible advantage to computing simplicity the document boundary is recommended if short texts are involved. Cross-sentence proximity boundaries are often absent in conventional Boolean systems and it does not appear to be a problem: the last word of one sentence and the beginning of the next are believed not often to comprise content-bearing words, and finding an example of a false match on this kind of adjacency for Ref. 9 was not easy. Sentence boundary recognition was very hard to automate on the test records in use, so this finding has welcome practical value. However, this may be appropriate only for short title and abstract records: for long texts the paragraph boundary may well be needed to avoid false matches.

Test 4: Stopwords not counted

Pairs proximity counts included stopwords by default, but as these were marked by an "$" in the record match profiles their non-counting could easily be tested. The same two proximity algorithms were used, and in both cases a very tiny performance degradation resulted from dropping the stopwords in the distance count and weights. Ranks dropped only to 25 from 22 in one case, and to 28 from 21 in the other: here there was a 4.5% drop in Precision at the middle range 50% Recall area.

The list of some 28 stopwords represented a medium strength approach to stopwording. Replication in other subject areas is again needed. Stopword counting would appear to be almost immaterial on these results.

Test 5: Every pairs test

Just one test of a simple count of every pair has been made, compared with the default of adjacent pairs and retaining the sentence and field boundaries in both cases. A full picture of the result appears in Table 4. Every pair degrades performance at the low and medium Recall range in the Precision Ratio, and at the medium and high Recall levels using the Output Position measure. Overall test rank drops from 34 to 46.

This is a little suprising, though the presence of the very repetitively indexed descriptor field in these records may well be the cause of this result. Future testing will need to examine this variable again and include some pairs distance testing rather than just counts of numbers of pairs.

Test 6: No descriptor field

The repetitive nature of the descriptor fields has been noted, and also the descriptor punctuation boundaries were ignored, so with a lack of stopwords in that field quite high phrase component occurrence and high weights were achieved. In case this were fuelling an exessive match with irrelevant items, three tests were done in which the descriptor field "sentence" was deleted. This was done using quorum and two sentences totalled algorithms. In all three cases a degradation in performance resulted, with drops of 5% merit for Quorum, and 13% and 22% for the Sentences algorithms. However, further work is needed to reduce the descriptor effect on matching, in case some milder modification proves beneficial.

Test 7: Phrases plus other term weighting combination tests

Ref. 5 reports the tests of four query term weighting methods, namely:

pairs distance product
collection frequency (=idf)
record frequency (=tf)
searcher weighted

The fifteen tests covered comparisons of each of the four single methods, and then all dual, triple and quadruple combinations. The form of the inverse collection frequency equation used was given in Ref. 3. Table 5 presents an example showing the way in which the four methods were implemented, particularly in combination. The first problem encountered was how to compute the pairs distance weights to be combinable with the other schemes which assigned weights only to each single term. The Table shows that the selection of five query terms provided five pairs matches, namely the term pairs "online/search", "search/academic", "service/library", "search/library" and "academic/library", with each pair having distance weights. It was decided simply to assign the full distance weights to each component term, just summing the weights where a term is the component of several pairs. Thus "online" was weighted 8, "search..." 24, because it had scores of 8, 7 and 9, and so on. These term distance weights were divided by a constant when used in some combinations, so that each scheme provided a very roughly similar order of magnitude of values: this correction was carried out using a constant for the whole retrieval test run of 35 queries matched against the 6004 records, and is illustrated by Table 5 having the distance weights divided by 10 and the three other weights not adjusted.

In one particular run using Pairs distance and Searcher Weighted, some imbalance in scores was suspected so a run involving reducing the Pairs score weight to 20% was made but it had almost no effect whatever on the performance, both scoring 54.3% merit and ranked 4. Another test of weight adjustment might appear to be more radical: the inverse collection frequency weights had values approaching 1 for very high frequency terms and a value of 13.55 for any term occurring once. The values were computed to two decimal places. In two tests the scale of the values was nearly halved, to run from 1 to 7, and integers were used. Runs of just inverse collection frequency on its own, and the perverse benchmark of direct collection frequency showed no disadvantage at being computed using reduced-range integers: merit ranks dropped from 40 to 42 and 67 to 68 respectively. Although this simplifying benefit was not applied in the combination runs, it seems likely that it would give quite acceptable results with a gain in the simplicity of integer operations. One further minor variant run was done using combination schemes: Proximate pairs and collection frequency were re-run with a small number of orphaned single terms added to the query/record match scores. This reduced the merit rank, trivially, from 13 to 15.

As Ref. 5 briefly discussed, combining the weights for each term/method can be done in many ways, and three methods were the subject of a preliminary selective test, listed as equations (a), (b) and (c) in Table 5. The Table shows the three methods in use, and Table 6 extracts from Table 1 the 26 test runs in which results from the three equations may be compared: some 11 cases of comparison were available, in nine of which the match method was held constant. Equation (b), sum before product, gave a slightly better result than sum, equation (a), in 5 of the 6 tests, and was better than equation (c) product before sum also in 5 of 6 cases, sometimes by a large amount. (b) was better than both (a) and (c) in 2 of the 4 cases, but was judged to be the best overall. It would be hard to choose a second best, but (a) would have computational simplicity in its favour.

Table 6 also extracts from Table 1 the fact that choices of term pair algorithm were explored in a small number of the many possible combinations with other schemes. In fact 5 choices were tested, 4 of the distance-incorporating methods plus the simple term pairs count. The pairs distance product algorithm which performed best on its own, as Table 3 has shown, also worked best as a combination component.

Table 7 reports the results of the final 15 comparisons, 11 of them the combinations, and gives more measures than Ref. 5 and orders the fifteen options by decreasing Merit and Rank. The combined methods occupy the top 8 of the 15 results.

Discussion

The first question to be discussed is how these new results compare in general with those of other schemes and other test collections, for example those obtained by Fagan and by Croft. Table 8 presents three of the new results with five of Fagan's and two of Croft's. The result of each scheme is shown as a simple difference (see Ref 4 for reasons why the percentage improvement is an unhelpful method) between that scheme and the result of the most sensible benchmark, that of collection and record frequency, normally known as idf x tf. Fagan's result on the CACM collection is best, and his tests on MED and CISI show the least improvement along with Croft's two results. The new LISA tests rank 2, 3 and 5 among the ten. It must be concluded that no dramatic performance improvement is yet in evidence.

The second question to be discussed is what the specification should be for a phrase or term proximity scheme in the light of the six variables tested here plus those expored by Fagan. As is indicated in Table 8, Fagan tested document boundaries and proximity levels, and found that three collections preferred no sentence boundaries but two did, and also unlimited proximity levels favoured four collections but in one case a level of one was best. In test 3 here, the LISA collection was best with no sentence boundary, thus suggesting that this be the sensible default for future work. Proximity levels are weighted in this research, so fixed levels are not advocated: the algorithms in test 2 showed adjacent pairs distance to be the best so far devised.

Test 5 followed Fagan's technique of recognising every possible pair, rather than just adjacent ones, and on LISA it bordered on being harmful, and should be rejected on grounds of implementation effort. Fagan's tests incorporated various document/term frequency thresholds, but Keen's results in Test 7 favour collection and record frequency schemes together with pairs distance by weighted functions and no thresholds. Ideas of trying to select only phrases judged 'good' by Fagan and schemes of query expansion by Croft are very effortful, and the present research looks for automatic methods to use instead: as an analysis revealed in Ref 2 sometimes a semantically invalid match improves the

Query Terms	Pairs Distance Weights	Distance /10	Record Term Weights Collection Frequency	Record Frequency	Searcher Weights	Product
online		.8	6.41	1	5	25.6
	8					
search...		2.4	4.12	6	2	118.6
	7					
service /s		.7	1.74	2	1	2.4
	7					
academic /s		1.7	4.98	2	1	16.9
	9					
librar /y /ies		2.6	1.61	4	1	16.7
	10					
	Totals	8.2	18.86	15	10	180.2

Equation (a) Sum of weights = 52.1
Equation (b) Sum of weights then product of types (/100) = 231.9
Equation (c) Product of term weights then sum = 180.2

Table 5: Selection of terms from query 32 as matched against record 400 showing three ways of combining all four term weighting schemes

Combination runs	Merit ranks Equations (a)	(b)	(c)
Pairs distance product* & Collection frequency	7	4*	NA
Distance within prox. pairs & Collection freq.	12	10	NA
Total distance & Collection frequency	34	--	27
Mean distance & Collection frequency	51	--	52
Proximate pairs & Collection frequency	29	--	13
Pairs distance product* & Record frequency	--	24*	--
Proximate pairs & Record frequency	--	--	59
Pairs distance product* & Searcher weighted	--	4*	--
Proximate pairs & Searcher weighted	--	--	23
Collection frequency & Record frequency	20	18*	16
Collection frequency & Searcher weighted	40	43*	55
Searcher weighted & Record frequency	48	30*	56
Collection f & Record f & Searcher w	16	14*	25

*Options used in runs in Table 7 and Ref. 5

Table 6: Merit ranks from Table 1 of the 26 tests of 3 combination equations and 14 tests of 5 term position algorithms.

Query Term Weighting Scheme	Merit	Rank	Low Recall Threshold 25%			Medium Recall Threshold 50%			High Recall Threshold 75%		
			Precision ratio	Output position		Precision ratio	Output position		Precision ratio	Output position	
	mn		mn	md	mn	mn	md	mn	mn	md	mn
Best possible	100%	1	100%	2	3.2	100%	4	5.7	100%	6	8.0
Pairs & Collectn & Record & Searcher	55.8%	2	71.4%	4	9.0	56.5%	12	22.0	39.4%	38	53.7
Pairs D & Collection & Record F	54.6%	3	68.4%	5	9.3	57.4%	11	23.0	37.9%	44	60.5
Pairs Distance & Collection Frequency	54.3%	4	70.6%	4	8.9	54.0%	12	22.9	38.2%	44	60.5
Pairs Distance & Searcher Weighted	54.3%	4	68.7%	4	8.4	54.7%	12	22.2	39.4%	32	68.7
Pairs D & Collection F & Searcher W	53.4%	9	68.4%	4	9.2	52.8%	13	22.4	39.1%	34	53.8
Pairs D & Searcher W & Record F	52.9%	11	68.4%	4	9.5	53.3%	12	25.2	37.0%	40	75.7
Collection F & Record F & Searcher W	51.6%	14	66.9%	5	11.3	50.9%	15	25.5	36.9%	50	55.6
Collection & Record Frequency	51.0%	18	65.3%	5	11.5	51.1%	16	26.9	36.8%	48	58.5
Pairs Distance	49.4%	21	62.4%	4	9.7	51.3%	14	26.7	34.5%	44	90.9
Pairs Distance & Record Frequency	48.3%	24	59.6%	6	11.9	52.0%	15	32.0	33.5%	53	95.2
Searcher Weighted & Record Frequency	46.7%	30	58.8%	7	14.0	47.1%	16	31.2	34.3%	47	72.7
Collection Frequency	43.0%	40	56.8%	5	19.4	42.5%	25	38.3	29.8%	53	91.3
Collection Frqncy & Searcher Weighted	42.7%	43	56.6%	5	16.7	41.7%	21	33.9	29.8%	57	78.3
Searcher Weighted	38.6%	57	49.1%	8	18.9	39.6%	22	37.5	27.0%	54	95.4
Record Frequency	33.1%	63	41.2%	10	29.5	33.2%	34	62.8	24.8%	79	122.4
Quorum	32.9%	64	45.4%	9	21.5	31.1%	28	44.3	22.0%	79	111.1

mn = mean md = median

Table 7: Test results for combination runs as Ref. 5 but adding mean Output position and ordering by Merit.

Difference %	Test Collection	Experimenter and scheme
5.9	CACM	Fagan, document domain, proximity unlimited
4.8	LISA	Keen, pairs & collection & record & searcher
3.6	LISA	Keen, pairs & collection & record
3.4	CRAN	Fagan, document domain, proximity unlimited
3.3	LISA	Keen, pairs & collection
2.9	INSPEC	Fagan, document domain, proximity unlimited
2.2	MED	Fagan, sentence domain, proximity unlimited
1.1	CACM	Croft, hybrid scheme
0.5	CISI	Fagan, sentence domain, proximity 1
-0.8	CACM	Croft, proximity scheme

Table 8: Ten results showing the performance difference compared with collection and record frequency (idf x tf), from Keen Ref 5 and this paper, Fagan Ref 8 and Croft Ref 7.

rank of relevant records.

Test 4 found stopword inclusion or exclusion for distance scores computation to be immaterial, and Test 6 concluded that all record fields should be included for pairs match purposes, even peculiarly structured controlled language fields. Ref 2 looked at term stemming, a technique practised quite strongly by Fangan and by Croft, but its value is called in question on the LISA tests, and avoiding even the use of s-stems did not degrade performance by more than a whisker. So, though it is a little premature to draw up any definitive scheme to incorporate phrases and proximity, the picture is a little clearer now.

Both Keen in Ref 1 and Croft in Ref 7 discuss phrases as a precision device, and this is a further question worthy of investigation. Refs 3 and 5 present one or two of the current combination scheme results as recall versus precision graphs, using document level cutoff values from 1 to 50. In both cases crossing performance curves were found, with schemes based just on collection or collection & record frequency being best at the high recall end, and pairs being best at the low recall high precision end. This does suggest that cleverly devised hybrids might be possible.

The question of comparing results such as these for ranked output retrieval with conventional Boolean searches was discussed in Ref 4, and some results using LISA appeared there and in Ref 9. The conclusions are tentative: ranked output can give a very good Precision at low Recall and also has the potential of reaching about 75% Recall, with Boolean having difficulties in these performance areas, but often giving a very good result at medium Recall. The theme of Boolean's confidence-enhancing feeling that 'the searcher is in control and knows what is happening and what can be done next' posited in Ref 5 needs some careful new investigations in user-simulated laboratory environments and later in operational ones. Also, as Ref 5 disusses, both the user and system overheads of ranked output schemes will need characterising more accurately.

A final research question is that of the suitablity of the laboratory test collections in use for experiments of this kind. Croft reports the move away from the CACM collection and Keen in Ref 3 sets out desiderata for collections, including fullness of records and queries, as well as topics which have troubled IR researchers for 30 years such as record and query sample sizes, relevance judgments, and so on. But there simply is no alternative to laboratory testing of as good a quality as is possible to keep chipping away at the problems of information retrieval. Now is not the time to stop.

References

1. KEEN, E. M. The use of term position devices in ranked output experiments. Journal of Documentation, Vol. 47 No. 1 March 1991 1-22

2. KEEN, E. M. The effect of stemming strength on the effectiveness of output ranking. In Informatics 11 Conference Proceedings, Aslib, 1991, 37-50

3. KEEN, E. M. The effectiveness of term position and frequency for output ranking. In Proceedings of the British Computer Society 13th Information Retrieval Colloquium, University of Lancaster, 1991, 22-37

4. KEEN, E. M. Presenting results of experimental retrieval comparisons. Paper to appear in Information Processing and Management Special issue on evaluation issues in information retrieval.

5. KEEN, E. M. Query term weighting schemes for effective ranked output retrieval. In Online Information 91, Proceedings of the 15th International Online Information Meeting December 1991, 135-142

6. SALTON, G and BUCKLEY, C. Term-weighting approaches in automatic text retrieval. Information Processing & Management, Vol. 24 No. 5 1988 513-523

7. CROFT, B., TURTLE, H. R. and LEWIS, D. D. The use of phrases and structured queries in information retrieval. Proceedings of the Fourteenth Annual International ACM/ SIGIR Conference on Research and Development in Information Retrieval, Chicago, 1991, 21-30

8. FAGAN, J. L. The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. Journal of the American Society for Information Science, Vol. 40 No. 2 March 1989 115-132

9. KEEN, E. M. Proximity searching in text retrieval systems. Paper given at the Text retrieval and in-house information systems conference, London, 6th-7th November, Institute of Information Scientists. Paper submitted to Journal of Information Science.

10. BETTS, R. and MARRABLE, D. Free text vs controlled vocabulary - retrieval precision and recall over large databases. In Online Information 91, Proceedings of the 15th International Online Information Meeting December 1991, 153-165