Decision Level Data Fusion for Routing of Documents in the TREC3 Context: A Best Case Analysis of Worst Case Results.

Paul B. Kantor
Department of Library and Information Studies and
Alexandria Project Laboratory (APLab), SCILS, Rutgers
kantor@kantor.rutgers.edu

ABSTRACT

The performance of a simulated test of decision level data fusion in the routing (filtering) task of the Text Retrieval Conference is summarized and analyzed. The relatively poor results of an approach in which a specific fusion rule was selected for each retrieval task are analyzed in terms of a best possible fusion scenario based on a given scheme for quantizing the messages from the systems to be combined. The limitations of that scenario are in turn explored, and possible ways to improve upon it are outlined.

I. INTRODUCTION

In the TREC3 setting, each participant submits two proposed ranked sets of retrieved documents for each of two sets of 50 problem or "topic" statements. The top portions of each retrieved set are pooled and evaluated for relevance (binary score) to the corresponding topic by trained evaluators. [See Harman (1995) for further details]. One set of topics, called the "Adhoc topics" are provided without any training information, and are to be run against a specified set of retrievable documents. The second set of topics, called the "Routing topics" a provided together with a set of relevance judgements for (selected items) from a training set of documents. This permits participants to "tune" the query formulations, retrieval systems, or fusion rules, before applying them to a "test" set of data. One expects that the performance of such tuned rules would be, in general, better than the performance of adhoc retrieval, and that the performance on the test set should be comparable to (but probably somewhat lower than) the performance on the training set.

The purpose of the present work is to simulate the situation in which two or more distinct systems are available, each providing evaluations of the "similarity score" (or "retrieval status value") for the same set of documents, to a given topic. In addition, we seek to simulate the situation in which the internal workings of the several systems and, in particular, the "scores" are hidden from view (either because they are proprietary, or because the working of the system e.g. a neural network, does not produce a numerical score). In this case each system provides only its ranked list of documents, for each topic. This is an instance of what is called "decision level" fusion. [Hull, Esp Ch 6]. That is, the ranked lists represent the decisions, made by the several systems, in response to the presented topic. In the work described here, a single very effective retrieval system, the University of Massachusetts Inquery system [Turtle and Croft] was used to perform all indexing, stemming and retrieval.

In an effort to simulate the case of several systems, three different retrieval modes of the Inquery system were used, and their results were treated as if they had come from three different systems. Two of these modes accept a Boolean formulation, and the third, called "natural language processing (nlp)" accepts strings of terms, with possible repetitions. Queries were formed by a "nearly algorithmic" process, carried out by one person, working without aid of thesaurus. The details of the fusion rules are set out below.

After the training set had been run, the apparently most effective fusion rule was selected for each routing topic.

Table 1. Overall Results of Data Fusion Trials. The measure used throughout is precision at 100 documents retrieved, averaged over 50 topics

Method	Precision at 100 Documents
rutfua1	.371
rutfua2	.359
rutfur1	.266
rutfur2	.267

File rutfur1 contains, for each topic, the results from that fusion scheme which did best on the training set for that topic. Best is determined by comparing precision at 100 documents. File rutfur2 contains, for each topic, the results from that fusion scheme which had second best performance on the training set for that topic. Best is determined as above.

The procedures for rutfur[1,2].test are exactly as described for the training set. The weights used are the ones determined from the training set. The searches were run at the University of Massachusetts, using Inquery, with (inverse) document frequencies determined on the training set only.

Since there were no training data for adhoc runs, the method which appeared to be most effective most often was applied to the adhoc topics. The adhoc runs, rutfua[1,2].test were run using the best data fusion schemes, based on overall performance on the training set. Specifically rutfua1.test uses a sum of ranks provided by the three component schemes, to determine an effective rank. rutfua2 uses the minimum of the ranks assigned by the three component schemes to determine an effective rank.

For the adhoc runs, rutfua1 and rutfua2 our scores fall near the median whichever measure of performance is used. As shown by Tague-Sutcliffe [Workshop presentation at TREC3], posthoc Scheffe tests can be applied to these data, treating the precision scores as the sum of an effect due to the topic and an effect due to the system. Under this analysis, using the precision at 100 documents as a score, the adhoc fusion results are not significantly worse [at 95% confidence] than the results obtained by using the full power of the Inquery system, which provided top performance among all the systems in TREC3. This lack of difference is presumably a reflection of the low power of the TREC setting to discriminate among systems, rather than any suggestion of parity between the two sets of results. A discussion of a somewhat less restrictive nonparametric approach to the comparison of systems, when their rank in the TREC setting is known, is given in an unpublished note [Kantor 1994].

For the routing case, as noted, a "tuned" choice of fusion rule was made for each topic. As a second set, the second best fusion rule was assigned for each topic. These results ranked dead last by several measures of system performance! The purpose of the present note is to lay out in more detail what was done, and to explore some details of the decision-level fusion process, in an effort to understand why the results were not better.

II. QUERY FORMULATION AND THE THREE SIMULATED SYSTEMS

The three simulated systems, which are to be combined in a variety of ways, begin with the reduction of the topic text to a Boolean expression as a single conjunction of disjuncts. This corresponds to the basic notion of "combination of concepts" as it is used in commercial Boolean set retrieval systems. The Boolean forms were constructed by a graduate student who added no vocabulary, and only made one correction to an error in the topic text. Proximity operators were used, but no weights were provided, as they would not be interpretable by all modes of the Inquery system. An example query is, for Topic #125:

```
Inference Form: #q125 =
#and(#or(government authority court)
     #or(law regulation control limit warning discourage funding research action)
     #or(#1(anti smoking) smoking tobacco)
     #not(#or(#2(price support) #2(export encouragement)))
);
```

This was submitted to the Inquery "inference mode" (inf), using the default settings for belief levels. [Turtle and Croft] This produces a ranked set output. It was also submitted to the "hard Boolean mode" (hbl) which produces a set retrieval without ranking.

The operators were removed to provide the query formulation for the "natural language processing mode" (nlp), using the default values for belief settings. This produces a ranked set. The Unstructured or "natural language processing" form for Topic 125 is:

```
#q125 =
government authority court law regulation control limit warning discourage funding research action anti smoking smoking tobacco ;
```

III. DECISION LEVEL DATA FUSION FOR INFORMATION RETRIEVAL.

Given the output of two or more systems or modes of a single system we combine them using several possible fusion rules. In order to illustrate the action of these rules we introduce a binning procedure which will be used for the remainder of this analysis. Such a binning procedure is also called a "quantization" of the decision-level signals issued by the systems. The set of ranks assigned by any mode is broken into 25 bins, each containing 40 consecutive ranks. These are labelled by the numbers 0,1, ... 24, which we call "bin ranks". A given document, if it is retrieved in the top 1000 by two different modes, will have two different bin ranks, (b_1,b_2). Hence it can be represented as appearing in the location indexed by that pair of coordinates.

We will also refer to those products of bins as bins, in the plane. In general there will be some number of documents, in each bin, which have been judged and judged relevant. We denote this by $g(b_1,b_2)$. Similarly, there will be some number that have been judged not relevant, which we denote by $b(b_1,b_2)$. The goal of any retrieval system faced with this information is to sweep across the plane of bins, in such a way that the accumulation of relevant documents is as rapid as possible, while the accumulation of not relevant documents is as slow as possible. This will be made more precise below.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2	2	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
3	3	3	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
4	4	4	4	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
5	5	5	5	5	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
6	6	6	6	6	6	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
7	7	7	7	7	7	7	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
8	8	8	8	8	8	8	8	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
9	9	9	9	9	9	9	9	9	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
10	10	10	10	10	10	10	10	10	10	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
11	11	11	11	11	11	11	11	11	11	11	11	12	13	14	15	16	17	18	19	20	21	22	23	24
12	12	12	12	12	12	12	12	12	12	12	12	12	13	14	15	16	17	18	19	20	21	22	23	24
13	13	13	13	13	13	13	13	13	13	13	13	13	13	14	15	16	17	18	19	20	21	22	23	24
14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	15	16	17	18	19	20	21	22	23	24
15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	16	17	18	19	20	21	22	23	24
16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	17	18	19	20	21	22	23	24
17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	18	19	20	21	22	23	24
18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	19	20	21	22	23	24
19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	20	21	22	23	24
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	21	22	23	24
21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	22	23	24
22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	23	24
23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	24
24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24

Exhibit **1**. Maximum logic for fusion of ranks. The combined rank of each bin is shown.

There are a number of a priori schemes which may be applied to pursue this goal. Three of the simplest are the symmetric rules: MAX, MIN and SUM. These may be defined formally by the rule which gives the combined bin rank b_{com} as a function of the values (b_1,b_2). These rules are shown in Table 2.

Table 2. Mathematical form of the rules of combination.	
Rule	Formula for combined rank
MIN	$\min(b_1,b_2)$
MAX	$\max(b_1,b_2)$
SUM	b_1+b_2

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
0	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
0	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
0	1	2	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
0	1	2	3	4	5	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
0	1	2	3	4	5	6	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
0	1	2	3	4	5	6	7	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
0	1	2	3	4	5	6	7	8	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
0	1	2	3	4	5	6	7	8	9	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
0	1	2	3	4	5	6	7	8	9	10	11	11	11	11	11	11	11	11	11	11	11	11	11	11
0	1	2	3	4	5	6	7	8	9	10	11	12	12	12	12	12	12	12	12	12	12	12	12	12
0	1	2	3	4	5	6	7	8	9	10	11	12	13	13	13	13	13	13	13	13	13	13	13	13
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	14	14	14	14	14	14	14	14	14	14
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15	15	15	15	15	15	15	15	15
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	16	16	16	16	16	16	16	16
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	17	17	17	17	17	17	17
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	18	18	18	18	18	18
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	19	19	19	19	19
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	20	20	20	20
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	21	21	21
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	22	22
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	23
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Exhibit **2**. Minimum logic for fusion of ranks. The combined rank of each bin is shown.

Each document is located in one of the bins, identified by the rank assigned to that bin by the inf system (reading down the page) and by the rank assigned to that bin by the nlp system (reading across the page). The number at the position of each bin indicates its rank under the fusion rule.

In Exhibits 1,2, and 3 we show the combined rank assigned to each bin in the plane, under each of the three rules of combination. In practice ties must be broken in some way, such as using one of the two bin coordinates as a second sort key. This is not material for the remainder of this analysis, and will not be pursued here.

It is clear from examination of Exhibit 1, that the MAXimum rule corresponds to the

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48

Exhibit **3**. Sum of ranks rule for combination of ranks. The combined rank of each bin is shown.

logical combination AND. That is, a bin will have combined rank less that, say, 5, if and only if the ranks assigned to it by both modes of retrieval are less than or equal to 5. Similarly, the MINimum rule corresponds to a logical OR. That is, a bin in the plane will have a rank less than or equal to 5 if either of the two modes assigns it a rank less than or equal to 5.

Finally, the SUM rule, while still treating the two coordinates symmetrically, does not correspond to a specific logic. Rather, it permits a tradeoff of high rank in one mode and low rank in another. There are many other possible symmetric rules of combination, which can be generated from basic symmetric functions of two or more variables as discussed by Kantor [1981]. The enumeration of symmetric logical rules for combining three inputs is given by Cherikh.

In our treatment of the routing problem we asked which of several rules provided the best performance, for a given topic, and then applied the same rule (using of course the same query formulations) to retrieval from the test set. In fact, since our experiments included fusion of decision level data (that is, ranked lists) from each of three modes of the Inquery system, we considered a fourth possible rule, MED which set the combined rank equal to the median of the three separate ranks. Our analysis, in this note, of the problems with tuning decision level fusion will be carried out using only two modes (the nlp and the inf).

IV. EXAMINING THE TEST AND TRAINING DATA TOGETHER

To get a better understanding of the problems that have arisen, we present the bin arrays for the test and training data on a single topic from the routing set, Topic 125. The data of interest are the numbers $g(b_1,b_2)$ and $b(b_1,b_2)$. However, the retrieval performance, by any standard measure, will be optimized if we sort the bins in decreasing value of the ratio:

$$r(b_1,b_2)=g(b_1,b_2)/b(b_1,b_2)$$

Since this may be undefined (if $b(b_1,b_2) =0$), we work instead with:

$$p(b_1,b_2)=g(b_1,b_2)/[g(b_1,b_2)+b(b_1,b_2)].$$

This is undefined only if there are no evaluated documents in the bin. In that case we represent the bin by a ".". To compress the ratio into a single digit we use a formula defining the graphic character, $c(b_1,b_2)$:

Table 3. Definition of Graphic Characters for Exhibit 4.	
$c(b_1,b_2)$	"." if g+b=0
	"!" if g>0 and b=0
	$Int(Log_2(1000*g/(g+b)))$

The largest value of the numerical character is $Int(Log_2(1000))=9$.

Table 4. Training data, for three modes of retrieval, for Topic 125.	
Inquery retrieval mode	Precision at 100 Documents
Hard Boolean	.02
Inferential	.46
Natural Language	.45

Examining the patterns in Exhibit 4, we see that the patterns look generally similar. This means, from the perspective of data fusion, that this should be a fairly good test case on which to explore the potential of data fusion. To do so we consider several possible ways in which to order the bins, in the test array.

One possibility is to pay no attention to the training set, and adopt an order such as one of the ones shown in Exhibits 1,2,3. A second possibility is to "learn" as much as possible from the training set, and to use the results of that learning to perform the data fusion on the test set. We first summarize the information that was available after the training run. Recall that, under the data fusion rules, the queries themselves are not to be modified on the basis of the retrieved set in the training run. All that can be changed is the rule of combination.

```
Topic 125
     Training Set                  Test Set
898.!.....!..............     998..!...................
8998798..!...............     9878.....................
!98.797........!.........     79878........!...........
!898.!8..8....!..........     .99987..9................
....78!!.8....8......!!..     !.8887.8..7..............
.!..!989.8...8!..........     ..88.98989..!............
..8.8.!!!889......8......     ...87!8!.8!.8............
.....!..8...!............     ...88..8788!..!.!........
!..8.....!....8..........     ...!8.!88.7....!.7.......
.!.8.!8.........!........     .......88..........8.....
..!!......!..............     ......88...8........!....
......!.......!........!.     ...8....8..!.............
...!...............!.....     ...........8.............
.....!...8...8...!.......     ............!.!..........
......!.!.....!........!.     .............!.......!.!.
........!.....!.....!....     ......................!..
....!....!...............     .........................
.........................     .........................
.........................     ......8.......!..........
.........................     .........................
.......................!.     .........................
.........................     .....................!...
..........8............!.     .........................
.........................     .........................
.........................     .......!.................
```

Exhibit **4**. Relative distributions of relevant and non-relevant documents in the training and test sets for Topic 125. Codes are defined in Table 3.

The precision, at 100 documents, for the original three runs of Topic 125 are shown in Table 4. These data show that our concentration on just two of the modes is certainly justified here, as the third (the hard Boolean form) seems to have very little to contribute. We note that this disparity might suggest that the problem with our performance is due to the insistence on symmetry among the three modes, which seems unreasonable in situations like the one shown in Table 4. However, the detailed results to follow show that the problems are deeper than that.

V. TUNING THE DATA FUSION RULE.

In the TREC context we explored the possibility of tuning, Topic by Topic, by considering each of several data fusion rules, and selecting, for each topic, the rule which performed best. We did not consider the question of whether differences among rules were statistically significant in making these choices. Four rules were considered: the maximum, the minimum, the equally weighted sum, and a weighted sum. Under the weighted sum rule the rank assigned to a document by each mode is divided by 0.02+p(100, Mode). The 0.02 offset is intended to prevent division by zero. p(100,Mode) is the precision of that Mode, at 100 documents, in retrieval from the test set. In the present case the quantized version of this formula becomes:

$$b_c=b_{hbl}/.02+b_{nlp}/.45+b_{inf}/.46.$$

Effectively, this gives the inf and nlp modes 22 or 23 times as much "weight", in the sense that after the first bin of the hbl retrieval is included, some 22 bins of each of the other two will be brought into the fusion result before the second bin from the bhl mode enters. The precise meaning of this is a somewhat unclear as the boolean system does not produce a ranked retrieval. Thus, at some point there are no more documents to be considered from that mode.

For the present, then, we consider just the inf and nlp modes, as shown in Exhibit 4. However, the performance of four possible rules of combination [for all three inputs] is shown in Table 5.

Table 5. Performance of rules of combination on the training set. Topic 125

MAX	MIN	MED	Sum(Wtd)	Sum(Equal)
.28	.30	.31	.05	.45

SUM(Equal) means the sum with equal weights. Based on these results, our first choice for the test situation should be the sum with equal weights. In our TREC runs, however, this alternative was not included. We do not know why the weighted sum performs so poorly for this topic or, for that matter, in general.

Given the disappointing results of the fusion rules selected here, we have explored the possibility of learning in much finer detail. That is, rather than sweeping across the array of bins in one of the three patterns shown in Exhibits 1,2,3, we consider picking and choosing among the specific bins in the plane, so as to produce the best possible results on the training set.

The most aggressive approach to this is to rank the bins in the plane in decreasing order of g/b. This produces the results shown in Figure 1. This figure contains a great deal of information on both the training and test performance. The horizontal axis records the number of non-relevant documents collected while sweeping across the bins in the indicated order.

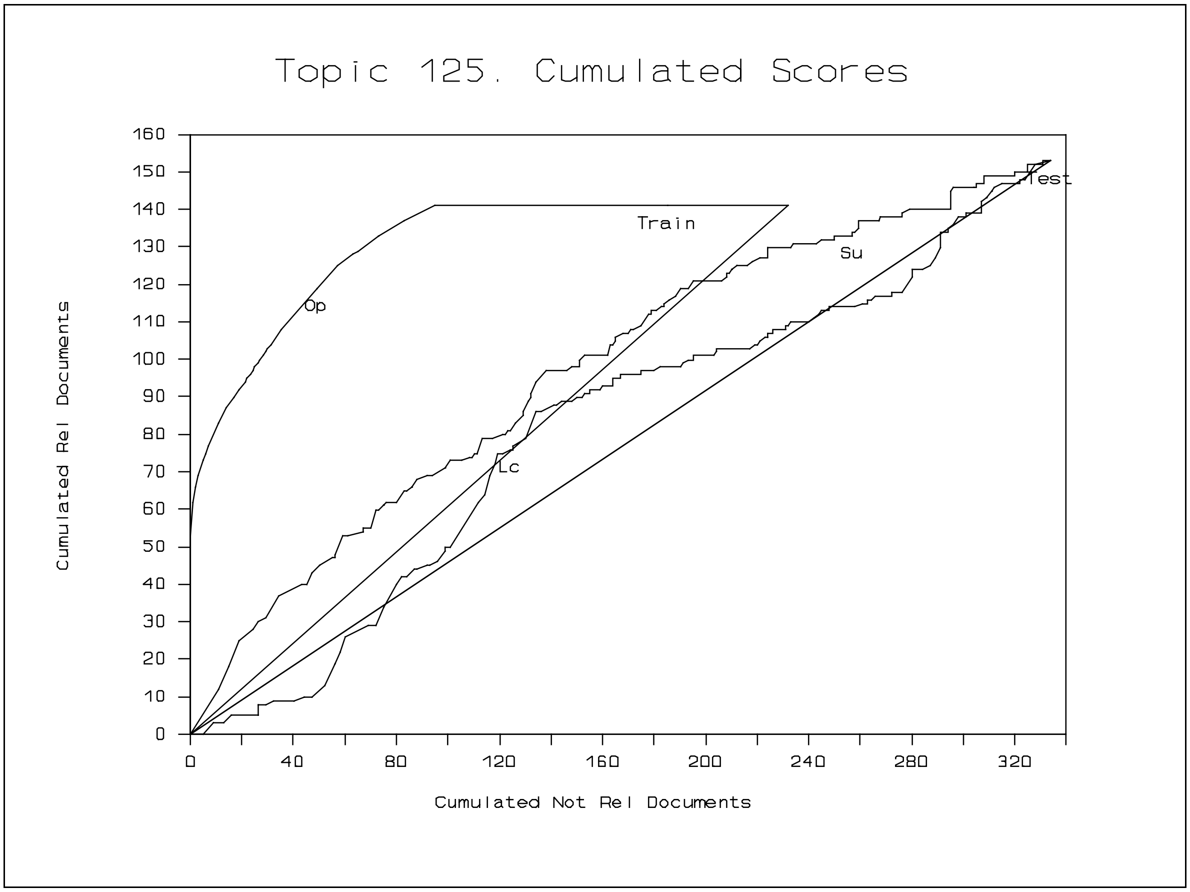

Figure 1. Cumulated Number of Good Documents. Bins sorted by g/(g+b).

The left-most curve , labelled "Op" is the cumulated curve for the training data if bins are collected in the optimal order (corresponding to first taking all bins marked "!", and then taking up the numbered bins in decreasing order, as shown in the left portion of Exhibit 4). The performance in this case, on the training set, is very good. To normalize this cumulated performance curve we have shown the straight line from the origin to the endpoint. This straight line represents the performance that would be achieved, on the average, if bins were simply taken in random order.

In the curves extending to the right-most portion of Figure 1, we show the corresponding structure for the test data. Again the straight line represents expected random performance. The jagged line labelled "Lc" (for "Local" tuning) shows the performance if the order of bin selection is exactly the same as the one used to generate the training curve. In the early portions it tracks the straight line, beginning below it, and rising above it. It lies above the random performance for an interval, and then falls below it again. Thus this highly detailed choice of tuning performs, overall, not very differently from a random selection rule. To calibrate the notion of "not very differently" we show, in the curve labelled "Su" the performance, on the training set, that is achieved by using the sum of ranks rule of Exhibit 3. This clearly everywhere dominates the performance of detailed local tuning. In other words, knowledge of the training set has not helped the data fusion at all!

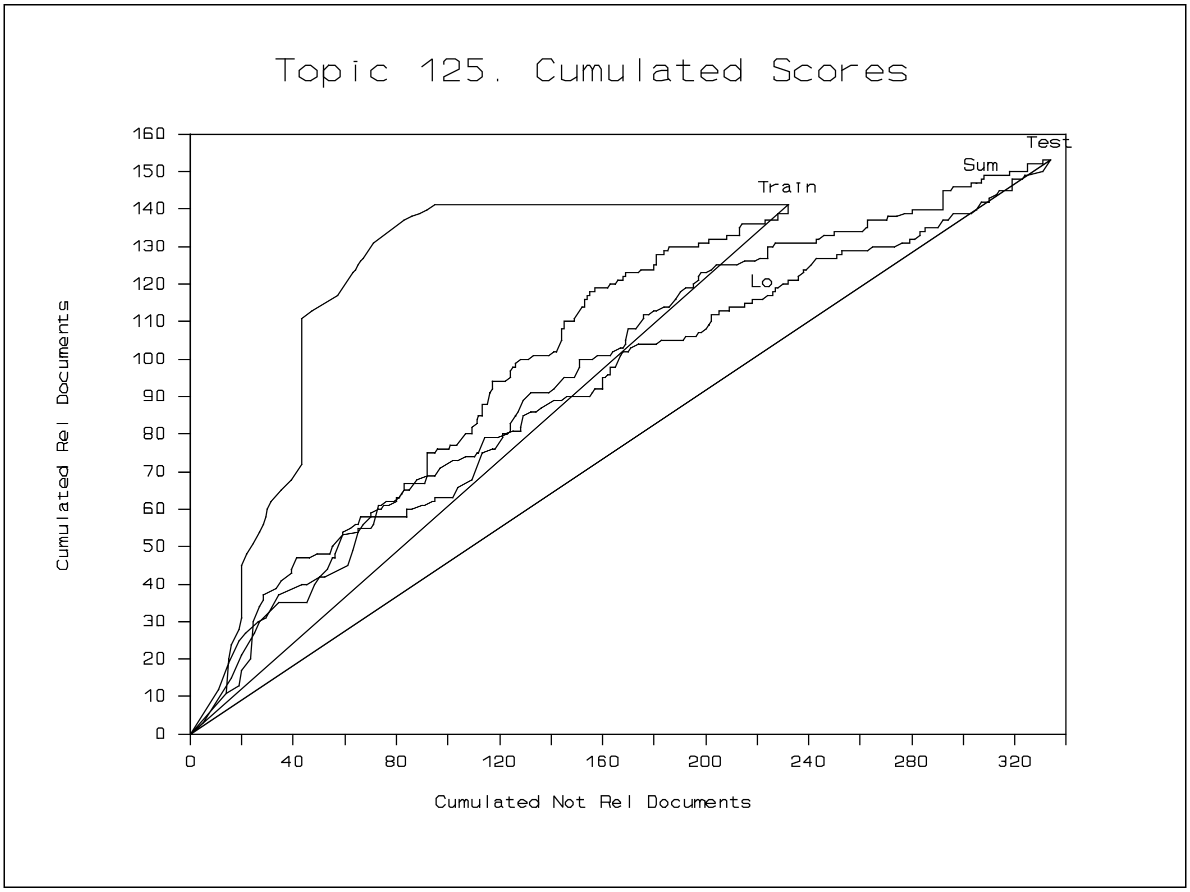

Figure 2. Cumluated relevant documents, using a rule which assigns greater weights to bins with more judged documents.

Table 6. A modified rule for determining combined rank of bins. "sqrt" represents the square root function.	
Situation	Combined Rank
b=0, g=0	-1
b=0, g>0	sqrt(g)
b>0, g=0	0
b>0, g>0	g/sqrt(g+b)

The obvious explanation for this is that we have "over trained". That is, we are using distinctions among bins which are due essentially to random fluctuations in the training data. Presumably these fluctuations are most prominent in bins containing small numbers of relevant (or non-relevant) documents. To test this, we have used another ranking scheme, which weights bins according to the square root of the total number of judged documents that they contain. The modified ranking rule is shown in Table 6.

The resulting cumulated curves of relevant versus non-relevant documents are shown in Figure 2. On the training set this does not perform as well as the optimal rule shown in Figure 1. But it does very well in comparison to the sum rule, shown as a jagged line rising from the origin to the endpoint of the training set ("Train"). On the test set, this less aggressive rule, labelled "Lo" is consistently better than random. But, except for a small region near the origin (which may well be due to fluctuations) it is still not better than the sum rule.

These results move us towards understanding why our application of data fusion to the routing situation did not work well. Disappointingly, they also suggest that even tuning that makes use of all the routing information will not do better than a symmetric fusion rule chosen a priori. However, the details of plots such as Exhibit 4 suggest that there may be other "generic" fusion rules which are more effective that the rules shown in Exhibits 1,2, and 3. In that plot we see that relevant documents are nicely clustered into the upper left corner, but, then, they continue across the plot in patterns that stay close to the main diagonal.

This suggests a rule for combination that does not correspond well to a simple logical expression. For example, it could be represented as $b_{com}=|b_{nlp}-b_{inf}|$. The vertical bars represent absolute value, and ensure that the combined rank is positive. The meaning of such rules is not yet clear. One possible interpretation is that when the two schemes agree on the rank that they assign, it is more likely that the document is relevant than that it is not. But why this should itself be true is not yet clear.

VI. DISCUSSION AND CONCLUSIONS

The clear conclusion of our efforts to select a "best symmetric tuning rule" for each of the routing topics is that, as implemented here, it does very poorly in the TREC setting. The first observation is that (Harman 1995) most TREC groups used results of the training data to improve their query formulations, eliminating terms that led to irrelevant documents, and adding terms that retrieved relevant documents. The evidence shows that this level of tuning is clearly superior to an approach that treats the query formulations as part of an impenetrable black box.

However, motivated by our observation that not all systems will permit their interiors to be manipulated in this way, we ask (perhaps over-optimistically) whether there are directions in which the present work might be extended to bring its performance closer to that of other training schemes.

We can represent the situation here using Q,R,S to represent systems, a prime (’) to represent the training of systems, F to represent fusion rules, and F’ to represent a trained fusion rule. Using < to mean "performs more poorly than" our present result is that F’(Q,R)<Q’. However, there is a body of evidence suggesting that F(Q,R)>Q, even if Q>R [Belkin, Kantor, Fox and Shaw].

One line of approach is to seek better formulations of the training rule F-->F'. This will require detailed exploration of schemes for defining and ordering the bins in the space of combined ranks. We have used prior fixed bins. Better results might be obtained, for example, by using a nearest neighbor scheme, or some other pattern analysis method [Fukunaga] to determine the ordering of the bins. The bins will have to be weighted along the lines used in the work shown in Figure 2, to avoid the effects of overtraining.

A second line of approach, to be investigated in future work, considers that perhaps our effort to simulate the case of different systems is not realistic enough. While we use different modes of the Inquery system, those modes still draw upon the same underlying stemming and indexing algorithms. Thus our "systems" Q,R,S are not very different from each other. [This is reflected, in Exhibit 4, by the tendency of the bins with high weight to lie close to the diagonal. This is a pattern which is not consistent with any of the rules shown in Exhibits 1,2,3.]. A very different approach is given by the so-called n-gram schemes [Cavnar, Damashek]. In direct evaluation [TREC-3] these schemes have not done as well as term-based schemes. But they are likely to be more nearly independent from those schemes, and hence to provide a more powerful basis for trainable data fusion schemes in information retrieval.

VII. ACKNOWLEDGEMENTS

Mr. Won-Sik Shim, of the Rutgers program in Library and Information Studies produced the "nearly automatic" but concept-structured query formulations. Mr James Callan processed them in several ways, using the Inquery software at the University of Massachusetts Laboratory for Information Retrieval Studies. We are indebted to Prof. W. Bruce Croft for this support as well. Prof. Nicholas Belkin at Rutgers installed and maintained the Inquery file file structure on the Rutgers computers, and provided the support of Mr. Richard Quatrain, of Electricite de France, who coded some of the post-processing data fusion scripts. The Alexandria Project Laboratory is supported by research funds from the US Department of Education, the Council on Library Resources, and other agencies. Additional support is provided by Dean Richard W. Budd of the School of Communication Information and Library Studies.

VIII. REFERENCES AND LITERATURE CITED

Belkin, NJ, Kantor, PB, Fox, EA, Shaw, JA. (to be published) Combining the Evidence of Multiple Query Representations for Information Retrieval. Information Processing and Management. [To be published].

Cavnar, WB. Using an N-Gram-Based Document Representation with a Vector Processing Model. In Harman 1995 op cit.

Cherikh, M. (1989) Optimal Decision and Detection in the Decentralized Case. Cleveland Ohio. Department of Operations Research, Case-Western Reserve University. PhD. Dissertation.

Damashek, M. Gauging Similarity via N-Grams: Language Independent Sorting, Characterization and Retrieval of Text. Prepritn: Fort George Meade, MD.

Fukunaga, K. Introduction to Statistical Pattern Recognition. Second Ed. Academic Press. 1990. 591pp.

Hall, DL. Mathematical Techniques in Multi-Sensor Data Fusion. Artech House. (1992).

Harmon, DK.[1995] The Third Text REtrieval Conference (TREC-3). NAtional Institutes of Standards and Technology. (This volume).

Kantor PB [1981]. The Logic of Weighted Queries. IEEE Transactions on Systems, Man and Cybernetics. v11(12)816-821. (1981).

Kantor PB [1984e]. A Simulation Method for Assessing the Significance of Non-Paramteric Comparisons in the TREC setting. Unpublished. Available by ftp from kantor.rutgers.edu.

Saracevic, T; Kantor PB. (1988) A Study of Information Seeking and Retrieving. III. Searchers, searches, overlap. Journal of the American Society for Information Science v39(3)197-216.

Turtle, HR; Croft, WB. (1991) Evaluation of an Inference network-based model. ACM Transactions on Information Systems. 9(3)187-222.